

**ITS**

Institut  
Teknologi  
Sepuluh Nopember

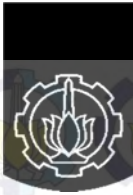
**TUGAS AKHIR - TE 145561**

**MODUL PENGATURAN KECEPATAN MOTOR SATU  
PHASA DENGAN PID BERBASIS MIKROKONTROLER  
MENGUNAKAN INTERFACE LABVIEW**

Ratih Putri Nastiti  
NRP 2213039033  
Wahyu Susanto  
NRP 2213039038

Dosen Pembimbing  
Suwito, S.T., M.T.  
Agus Suhanto, S.Pd.

PROGRAM STUDI D3 TEKNIK ELEKTRO  
Fakultas Teknologi Industri  
Institut Teknologi Sepuluh Nopember  
Surabaya 2016



**ITS**

Institut  
Teknologi  
Sepuluh Nopember

**FINAL PROJECT - TE 145561**

## **SPEED CONTROL MODULE WITH PID SINGLE PHASE MICROCONTROLLER INTERFACE USING LABVIEW**

Ratih Putri Nastiti  
NRP 2213039033  
Wahyu Susanto  
NRP 2213039038

Advisor  
Suwito, S.T., M.T.  
Agus Suhanto, S.Pd.

PROGRAM STUDI D3 TEKNIK ELEKTRO  
Fakultas Teknologi Industri  
Institut Teknologi Sepuluh Nopember  
Surabaya 2016

## PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul "**Modul Pengaturan Kecepatan Motor Satu Fasa Dengan PID Berbasis Mikrokontroler Menggunakan Interface LabVIEW**" adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

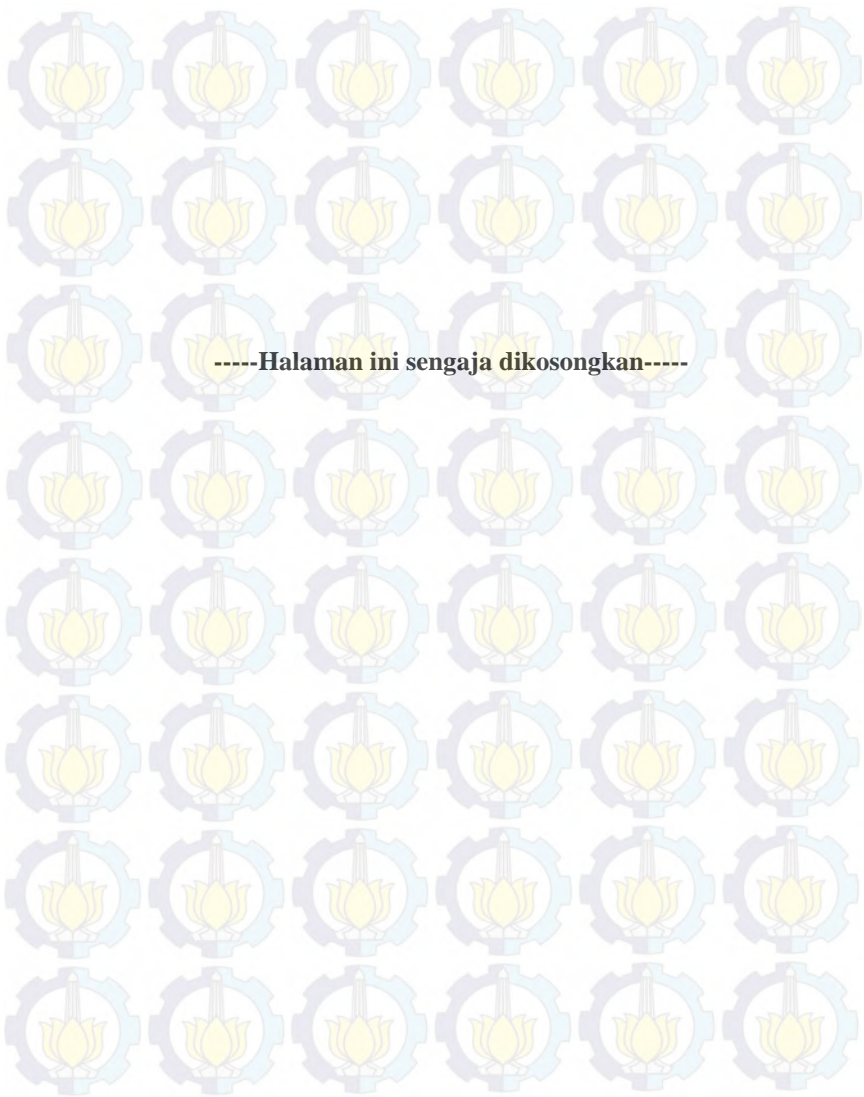
Surabaya, 1 Juni 2016



Ratih Putri Nastiti  
2213039033



Wahyu Susanto  
2213039038





**MODUL PENGATURAN KECEPATAN MOTOR SATU PHASA  
DENGAN PID BERBASIS MIKROKONTROLER  
MENGUNAKAN *INTERFACE* LABVIEW**

**TUGAS AKHIR**

Diajukan Guna Memenuhi Sebagian Persyaratan  
Untuk Memperoleh Gelar Ahli Madya Teknik  
Pada

Bidang Studi Elektro Industri  
Program Studi D3 Teknik Elektro  
Jurusan Teknik Elektro  
Institut Teknologi Sepuluh Nopember

Menyetujui:

Dosen Pembimbing I

Dosen Pembimbing II

Suwito, ST., MT.

NIP. 19810105 200501 1 004

Agus Suhanto, S.Pd.

NIP. 19650821 198603 1 010

**SURABAYA  
JUNI, 2016**



# **MODUL PENGATURAN KECEPATAN MOTOR SATU PHASA DENGAN PID BERBASIS MIKROKONTROLLER MENGUNAKAN *INTERFACE* LABVIEW**

**Nama Mahasiswa** : Ratih Putri Nastiti  
**NRP** : 2213 039 033  
**Nama Mahasiswa** : Wahyu Susanto  
**NRP** : 2213 039 038  
**Dosen Pembimbing 1** : Suwito, S.T., M.T.  
**NIP** : 19810105 200501 1 004  
**Dosen Pembimbing 2** : Agus Suhanto, S.Pd.  
**NIP** : 19650821 198603 1 010

## **ABSTRAK**

Motor induksi merupakan motor listrik arus bolak-balik yang banyak penggunaannya. Kelemahan motor induksi saat terjadi perubahan beban tidak mampu mempertahankan kecepatannya.

Pada Tugas Akhir ini diuraikan sistem pengaturan kecepatan motor induksi satu phasa menggunakan metode PID untuk menghasilkan kecepatan motor sesuai dengan *setpoint* saat motor tanpa beban atau berbeban. Pengaturan kecepatan ini dikendalikan dengan tegangan jala-jala 220VAC diubah menjadi tegangan DC diubah kembali menjadi tegangan AC dengan frekuensi yang sesuai menggunakan metode *switching* yang dikendalikan oleh mikrokontroler.

Dengan adanya sistem ini, motor dengan sistem *open loop* menggunakan *setpoint* 2500 rpm tanpa beban didapatkan nilai *rise time* 30 detik. Ketika motor diberi beban, kecepatan motor turun hingga motor berhenti. Sedangkan motor pada sistem *close loop* dengan PID menggunakan *setpoint* yang sama didapatkan nilai *rise time* 1,2 detik. Kecepatan motor turun ketika diberi beban, akan tetapi dapat kembali ke nilai *setpoint*-nya dengan nilai *rise time* 4 detik. Hal ini dikarenakan pada sistem *open loop* tidak terdapat umpan balik sedangkan pada sistem *close loop* dengan PID terdapat umpan balik berupa nilai kecepatan motor.

**Kata Kunci** : PID, LabVIEW, *Rise Time*





# ***SPEED CONTROL MODULE WITH PID SINGLE PHASE MICROCONTROLLER INTERFACE USING LABVIEW***

**Student Name's** : Ratih Putri Nastiti  
**NRP** : 2213 039 033  
**Student Name's** : Wahyu Susanto  
**NRP** : 2213 039 038  
**Advisor 1** : Suwito, S.T., M.T.  
**NIP** : 19810105 200501 1 004  
**Advisor 2** : Agus Suhanto, S.Pd.  
**NIP** : 19650821 198603 1 010

## ***ABSTRACT***

*Induction motors are electric motors of alternating current that much use. Induction motor weakness occurs when the load changes are not able to maintain his speed.*

*In this Final Project described induction motor speed control system using PID single phase to produce in accordance with the motor speed setpoint when the motor no-load or load. The speed setting is controlled by the voltage of 220VAC grid is converted into DC voltage is converted back into AC voltage with a frequency appropriate to use the method of switching which is controlled by the microcontroller.*

*With this system, the motor with open-loop system using 2500 rpm with no load setpoint values obtained rise time of 30 seconds. When the motor is loaded, the motor speed down until the motor stops. While the motor close loop system with PID using the same setpoint values obtained rise time of 1.2 seconds. Motor speed drops when it is loaded, but can be returned to its setpoint value with a rise time value of 4 seconds. This is because the open-loop system there is no feedback while in close loop system with PID are feedback of motor speed value.*

***Keywords : PID, Rotary Encoder, Rise Time***



## KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Tuhan Yang Maha Esa yang selalu memberikan rahmat dan hidayah-Nya sehingga Tugas Akhir ini dapat terselesaikan dengan baik

Tugas Akhir ini disusun untuk memenuhi sebagian persyaratan guna menyelesaikan pendidikan Diploma pada Bidang Studi Elektro Industri, Program Studi D3 Teknik Elektro, Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember Surabaya dengan judul :

### **" MODUL PENGATURAN KECEPATAN MOTOR SATU PHASA DENGAN PID BERBASIS MIKROKONTROLER MENGGUNAKAN *INTERFACE* LABVIEW"**

Dengan terselesaikannya Tugas Akhir ini, Penulis menyampaikan terima kasih yang sebesar - besarnya kepada :

1. Kedua orang tua yang senantiasa mendo'akan dan memberikan dukungan dengan tulus tiada henti.
2. Bapak Suwito, S.T., M.T. selaku dosen pembimbing.
3. Bapak Agus Suhanto, S.Pd selaku dosen pembimbing dari BLKIP.
4. Teman – teman D3 Teknik Elektro Angkatan 2013 yang selalu memberikan doa, semangat, dan dukungannya.
5. Semua pihak yang telah membantu baik secara langsung maupun tidak langsung dalam proses penyelesaian Tugas Akhir ini.

Penulis menyadari dan memohon maaf atas segala kekurangan pada Tugas Akhir ini. Akhir kata, semoga Tugas Akhir ini dapat bermanfaat dalam pengembangan keilmuan di kemudian hari.

Surabaya, Juni 2016

Penulis





## DAFTAR ISI

<b>JUDUL</b> .....	i
<b>PERNYATAAN KEASLIAN TUGAS AKHIR</b> .....	v
<b>LEMBAR PENGESAHAN</b> .....	vii
<b>ABSTRAK</b> .....	ix
<b>ABSTRACT</b> .....	xi
<b>KATA PENGANTAR</b> .....	xiii
<b>DAFTAR ISI</b> .....	xv
<b>DAFTAR GAMBAR</b> .....	xvii
<b>DAFTAR TABEL</b> .....	xix

### **BAB I PENDAHULUAN**

1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	1
1.3 Batasan Masalah .....	1
1.4 Tujuan .....	2
1.5 Metodologi .....	2
1.6 Sistematika .....	2
1.7 Relevansi .....	3

### **BAB II TEORI PENUNJANG**

2.1 Motor AC 1 Fasa .....	5
2.2 <i>Zero Crossing Detector</i> .....	8
2.3 TRIAC .....	9
2.4 <i>Incremental Rotary Encoder</i> .....	11
2.5 DAC 0808 .....	12
2.6 Arduino UNO .....	13
2.7 Kontrol PID .....	15
2.8 LabVIEW .....	22

### **BAB III PERANCANGAN DAN PEMBUATAN ALAT**

3.1 Blok Fungsional Sistem .....	25
3.2 Perancangan Perangkat Keras ( <i>Hardware</i> ) .....	26
3.2.1 Perancangan Elektronik .....	26
3.2.1.1 Perancangan <i>Power Supply</i> .....	26
3.2.1.2 Skematik Arduino UNO (1) .....	27
3.2.1.3 Skematik Arduino UNO (2) .....	28
3.2.1.4 Perancangan DAC 8 bit .....	29

3.2.1.5	Perancangan <i>Zero Crossing Detector</i> ..	31
3.2.1.6	Perancangan <i>Driver TRIAC</i> .....	32
3.2.1.7	Perancangan Sensor <i>Rotary Encoder</i> ...	32
3.2.2	Perancangan Mekanik .....	33
3.2.2.1	Perancangan Kopel Sensor .....	33
3.2.2.2	Perancangan Rem Mekanik .....	34
3.3	Perancangan Perangkat Lunak ( <i>Software</i> ) .....	34
3.3.1	Pemrograman Arduino UNO .....	34
3.3.1.1	Program Pembaca Kecepatan <i>Rotary Encoder</i> .....	35
3.3.1.2	Program Komunikasi Serial .....	37
3.3.1.3	Program PID .....	37
3.3.1.4	Program Konversi Desimal Menjadi Biner .....	39
3.3.1.5	Program Pengendali <i>Driver TRIAC</i> .....	40
3.3.2	Pemrograman LabVIEW .....	40
<b>BAB IV PENGUJIAN DAN ANALISA DATA</b>		
4.1	Pengujian <i>Power Supply</i> .....	43
4.2	Pengujian DAC 8 bit .....	43
4.3	Pengujian <i>Zero Crossing Detector</i> .....	44
4.4	Pengujian <i>Driver TRIAC</i> .....	45
4.5	Pengujian <i>Rotary Encoder</i> dan Program Pembaca .....	46
4.6	Pengujian Kontrol PID .....	48
<b>BAB V PENUTUP</b>		
5.1	Kesimpulan .....	59
5.2	Saran .....	59
<b>DAFTAR PUSTAKA</b> .....		61
<b>LAMPIRAN A FOTO ALAT</b> .....		A-1
<b>LAMPIRAN B PROGRAM</b> .....		B-1
<b>LAMPIRAN C DATASHEET</b> .....		C-1
<b>DAFTAR RIWAYAT HIDUP</b> .....		D-1

## DAFTAR GAMBAR

<b>Gambar 2.1</b>	Konstruksi Umum Motor Induksi Satu Phasa .....	6
<b>Gambar 2.2</b>	Karakteristik Pengaturan Kecepatan Terhadap Perubahan Frekuensi .....	8
<b>Gambar 2.3</b>	Karakteristik Perubahan Tegangan Terhadap Kecepatan Motor Induksi .....	8
<b>Gambar 2.4</b>	Grafik Tegangan <i>Input</i> dan Persilangan Nol	9
<b>Gambar 2.5</b>	Gelombang <i>Input</i> dan <i>Output Zero Crossing</i>	9
<b>Gambar 2.6</b>	Struktur dan Simbol TRIAC.....	10
<b>Gambar 2.7</b>	Daerah Kerja TRIAC.....	11
<b>Gambar 2.8</b>	Tiga Sinyal Keluaran <i>Encoder</i> .....	12
<b>Gambar 2.9</b>	Konfigurasi Pin IC DAC 0808 .....	13
<b>Gambar 2.10</b>	Blok Diagram Kontrol PID .....	16
<b>Gambar 2.11</b>	Waktu Respon Sistem .....	17
<b>Gambar 2.12</b>	Grafik Parameter <i>Plant K</i> , <i>Td</i> , dan <i>Ti</i> .....	18
<b>Gambar 2.13</b>	Pengaturan Sistem <i>Loop</i> Terbuka Ziegler-Nichols .....	18
<b>Gambar 2.14</b>	<i>Continuous-time</i> Sistem Kontrol PID .....	20
<b>Gambar 2.15</b>	Kontrol PID Dalam Struktur Paralel .....	20
<b>Gambar 2.16</b>	Respon $T_{ou}$ .....	21
<b>Gambar 3.1</b>	Blok Fungsional Sistem .....	26
<b>Gambar 3.2</b>	Rangkaian <i>Power Supply</i> .....	27
<b>Gambar 3.3</b>	Skematik Arduino Uno (1) .....	28
<b>Gambar 3.4</b>	Skematik Arduino Uno (2) .....	29
<b>Gambar 3.5</b>	Skematik DAC0808 .....	30
<b>Gambar 3.6</b>	Pengkonversi Arus Menjadi Tegangan .....	30
<b>Gambar 3.7</b>	Rangkaian Keseluruhan DAC 8bit .....	30
<b>Gambar 3.8</b>	Rangkaian <i>Zero Crossing Detector</i> .....	31
<b>Gambar 3.9</b>	<i>Driver</i> TRIAC AC 1 fasa .....	32
<b>Gambar 3.10</b>	Rangkaian MOC3021 .....	32
<b>Gambar 3.11</b>	Sensor <i>Rotary Encoder</i> .....	33
<b>Gambar 3.12</b>	Kopel Sensor .....	34
<b>Gambar 3.13</b>	Perancangan Rem Mekanik .....	34
<b>Gambar 3.14</b>	<i>Flowchart</i> program .....	35
<b>Gambar 3.15</b>	Program Arduino Pembaca Kecepatan .....	36
<b>Gambar 3.16</b>	Program Dasar Komunikasi Serial .....	37
<b>Gambar 3.17</b>	Blok Diagram Parameter PID pada Program.....	38
<b>Gambar 3.18</b>	Program PID .....	39

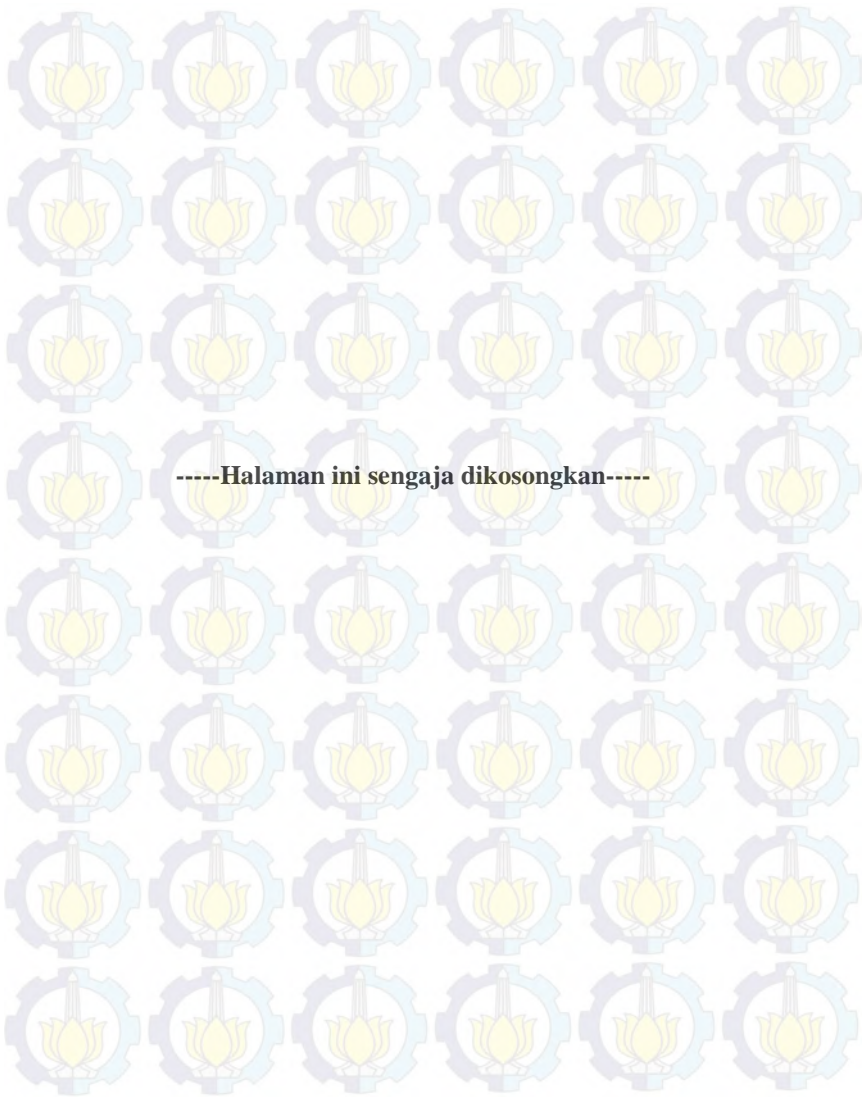


<b>Gambar 3.19</b>	Program Konversi Desimal Menjadi Biner..	40
<b>Gambar 3.20</b>	Program Pengendali <i>Driver</i> TRIAC .....	40
<b>Gambar 3.21</b>	Blok Diagram Pemrograman LabVIEW .....	41
<b>Gambar 3.22</b>	<i>Front Panel</i> Pemrograman LabVIEW .....	41
<b>Gambar 4.1</b>	<i>Wiring</i> Pengujian DAC 8 bit .....	44
<b>Gambar 4.2</b>	<i>Wiring</i> Pengujian <i>Zero Crossing Detector</i> ..	45
<b>Gambar 4.3</b>	Hasil Pengujian <i>Zero Crossing Detector</i> .....	45
<b>Gambar 4.4</b>	Pengujian Keluaran <i>Rotary Encoder</i> .....	48
<b>Gambar 4.5</b>	Pengujian Program Pembacaan <i>Rotary Encoder</i> .....	48
<b>Gambar 4.6</b>	Grafik Respon Motor Sistem <i>Open Loop</i> Tanpa Beban .....	49
<b>Gambar 4.7</b>	Grafik Respon Motor Sistem <i>Open Loop</i> Berbeban .....	49
<b>Gambar 4.8</b>	Grafik Respon Motor pada Nilai $K_p$ 1,7 .....	51
<b>Gambar 4.9</b>	Grafik Respon Motor pada Nilai $K_p$ 2 .....	52
<b>Gambar 4.10</b>	Grafik Respon Motor pada Nilai $K_p$ 2,3 .....	52
<b>Gambar 4.11</b>	Grafik Respon Motor pada Nilai $K_i$ 1 .....	53
<b>Gambar 4.12</b>	Grafik Respon Motor pada Nilai $K_i$ 1,7 .....	53
<b>Gambar 4.13</b>	Grafik Respon Motor pada Nilai $K_i$ 2,2 .....	54
<b>Gambar 4.14</b>	Grafik Respon Motor pada Nilai $K_d$ 0,1 .....	54
<b>Gambar 4.15</b>	Grafik Respon Motor pada Nilai $K_d$ 0,5 .....	55
<b>Gambar 4.16</b>	Grafik Respon Motor pada Nilai $K_d$ 1 .....	55
<b>Gambar 4.17</b>	Grafik Perbandingan <i>Rise Time</i> Terhadap <i>Setpoint</i> .....	56
<b>Gambar 4.18</b>	Grafik Perbandingan <i>Settling Time</i> Saat Beban Naik dan Turun .....	57



## DAFTAR TABEL

<b>Tabel 2.1</b>	Tabel Spesifikasi Arduino UNO .....	14
<b>Tabel 3.1</b>	Pin Arduino UNO (1) yang Digunakan .....	27
<b>Tabel 3.2</b>	Pin Arduino UNO (2) yang Digunakan .....	28
<b>Tabel 4.1</b>	Hasil Pengujian <i>Power Supply</i> .....	43
<b>Tabel 4.2</b>	Hasil Pengujian DAC 8 bit .....	44
<b>Tabel 4.3</b>	Hasil Pengujian <i>Driver</i> TRIAC .....	46
<b>Tabel 4.4</b>	Pengujian Program Pembacaan <i>Rotary Encoder</i> ....	47
<b>Tabel 4.5</b>	Pengujian Nilai Kp .....	50
<b>Tabel 4.6</b>	Pengujian Nilai Kp dan Ki .....	50
<b>Tabel 4.7</b>	Pengujian Nilai Kp, Ki, dan Kd .....	51
<b>Tabel 4.8</b>	Pengujian <i>Rise Time</i> Respon Motor Tanpa Beban ..	56
<b>Tabel 4.9</b>	Pengujian <i>Steady State</i> Berbeban .....	57



# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Motor induksi satu phasa digunakan secara luas pada berbagai aplikasi industri kecil dan rumah tangga karena harga yang murah, handal dan kuat. Sumber tegangan tiga phasa umumnya tidak tersedia pada pelanggan rumah tangga, oleh karena itu motor induksi satu phasa merupakan pilihan utama karena dapat langsung dihubungkan dengan sumber tegangan AC.

Kecepatan putaran motor induksi satu phasa dipengaruhi oleh frekuensi, jumlah kutub, tegangan jala-jala dan beban mekanisnya. Pada keadaan tanpa beban, kecepatan putaran motor induksi mendekati kecepatan medan magnet putar (kecepatan sinkron). Bila motor dibebani, kecepatan putaran motor berkurang akibat adanya torka beban yang menyebabkan slip pada motor induksi.

Dalam pengoperasian motor induksi satu phasa, kecepatan putaran motor tidak konstan. Kecepatan maksimum motor terjadi ketika motor tidak dibebani. Kecepatan motor jauh berkurang karena adanya beban yang harus dipikul oleh motor. Sedangkan, pada sebuah industri diperlukan kecepatan putaran mesin yang dapat diatur dengan tetap mempertahankan kestabilan putaran. Hal ini mengindikasikan bahwa kecepatan putaran yang diperlukan adalah kecepatan dimana motor sedang mendapat beban penuh. Pengaturan tegangan yang masuk ke motor dapat mengubah kecepatan motor mengikuti nilai tegangan yang disalurkan pada motor AC satu phasa yang digunakan.

### **1.2 Rumusan Masalah**

Permasalahan yang dibahas dalam Tugas Akhir ini adalah pengaturan kecepatan motor AC satu phasa dengan beban atau tanpa beban menggunakan kontrol PID

### **1.3 Batasan Masalah**

Dari perumusan masalah di atas, maka batasan masalah dari Tugas Akhir ini adalah kecepatan motor yang diatur oleh sudut penyalan yang dikendalikan melalui program Arduino UNO dan LabVIEW menggunakan kontrol PID.

#### 1.4 Tujuan

Pembuatan Tugas Akhir ini bertujuan untuk merancang dan membuat alat untuk mengontrol sudut penyalan TRIAC dalam pengaturan kecepatan yang stabil pada motor induksi satu fasa dengan beban atau tanpa beban menggunakan kontrol PID

#### 1.5 Metodologi

Penelitian ini dilakukan melalui beberapa tahapan metodologi, yaitu, studi literatur, pemodelan sistem, simulasi hasil desain, implementasi dan analisis data, dan yang terakhir adalah penyusunan laporan berupa buku Tugas Akhir.

Pada tahap studi literatur, dilakukan pencarian beberapa dasar-dasar teori yang mendukung pengerjaan Tugas Akhir yaitu sensor *rotary encoder*, rangkaian ZCD, rangkaian *driver* TRIAC dan juga penggunaan *software* Arduino UNO dan LabVIEW. Pengumpulan data diperlukan untuk mengetahui kecepatan motor AC induksi satu fasa dan respon PID yang dihasilkan dari motor tersebut. Pemodelan sistem dimulai dari penyusunan *flowchart* sistem alat, perancangan desain 3D alat, lalu pembelian komponen yang diperlukan. Setelah pembelian, diperlukan suatu pembuatan dari rangkaian-rangkaian yang akan disambungkan ke motor. Kemudian dilakukan *assembly* untuk proses perakitan dari alat yang meliputi proses perakitan kerangka dan juga proses pemasangan komponen berdasarkan desain yang telah dirancang. Untuk mengetahui apakah alat dapat mencapai target atau tidak, dilakukan simulasi alat dan mengumpulkan data-data yang diperlukan (respon PID, tegangan dan kecepatan akhir) untuk dianalisis lebih lanjut. Data-data dan hasil analisis tersebut dikumpulkan menjadi Tugas Akhir untuk implementasi lebih lanjut.

#### 1.6 Sistematika

Sistematika pembahasan Tugas Akhir ini terdiri dari lima bab, yaitu Pendahuluan, Teori Penunjang, Perancangan dan Pembuatan Alat, Pengujian dan Analisa Alat, serta Penutup.



## **BAB I : PENDAHULUAN**

Membahas tentang latar belakang, permasalahan, batasan masalah, maksud dan tujuan, sistematika laporan, serta relevansi.

## **BAB II : TEORI PENUNJANG**

Membahas tentang teori - teori penunjang yang diperlukan dan dipergunakan sebagai penunjang pengerjaan Tugas Akhir.

## **BAB III : PERANCANGAN DAN PEMBUATAN ALAT**

Membahas tentang perencanaan dan pembuatan perangkat keras (*hardware*) yang terdiri dari perancangan elektronik dan perancangan mekanik serta pembuatan dan perancangan perangkat lunak (*software*).

## **BAB IV : PENGUJIAN DAN ANALISA ALAT**

Membahas tentang pengujian alat yang terdiri dan pengujian perangkat keras dan juga perangkat lunak. Begitu pula dengan pengukuran.

## **BAB V : PENUTUP**

Menjelaskan tentang kesimpulan dari Tugas Akhir ini dan saran-saran untuk pengembangan alat ini lebih lanjut.

### **1.7 Relevansi**

- Mendapatkan kecepatan yang sesuai dengan yang diinginkan (*set point*) dari sebuah motor induksi satu fasa
- Motor induksi satu fasa memiliki kecepatan yang konstan walaupun mendapat perubahan beban.



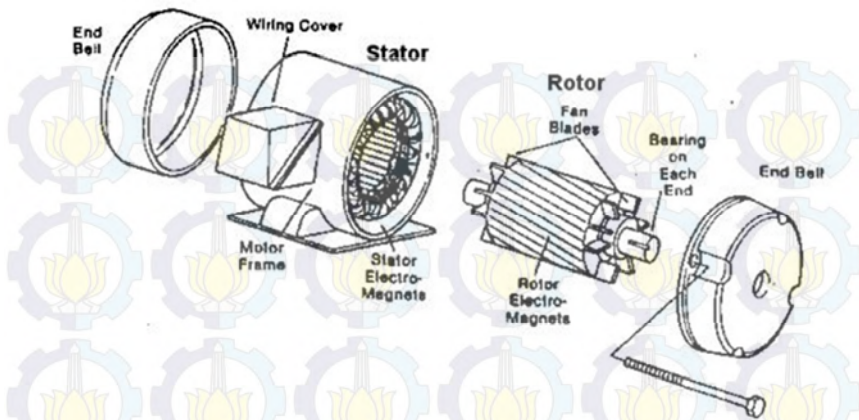
## **BAB II**

### **TEORI PENUNJANG**

Pada bab ini dibahas mengenai teori - teori yang menunjang dalam proses pembuatan alat Tugas Akhir. Teori yang menunjang dalam Tugas Akhir ini diantaranya adalah motor AC 1 fasa, Arduino UNO, *incremental rotary encoder*, PID, rangkaian DAC, LabVIEW.

#### **2.1 Motor AC 1 Fasa [1]**

Motor induksi satu fasa sering digunakan sebagai penggerak pada peralatan yang memerlukan daya rendah dan kecepatan yang relatif konstan. Hal ini disebabkan karena motor induksi satu fasa memiliki beberapa kelebihan yaitu konstruksi yang cukup sederhana, kecepatan putar yang hampir konstan terhadap perubahan beban, dan umumnya digunakan pada sumber jala-jala satu fasa yang banyak terdapat pada peralatan domestik. Walaupun demikian motor ini juga memiliki beberapa kekurangan, yaitu kapasitas pembebanan yang relatif rendah, tidak dapat melakukan pengasutan sendiri tanpa pertolongan alat bantu dan efisiensi yang rendah. Konstruksi motor induksi satu fasa hampir sama dengan konstruksi motor induksi tiga fasa, yaitu terdiri dari dua bagian utama yaitu stator dan rotor seperti yang terlihat pada Gambar 2.1. Keduanya merupakan rangkaian magnetik yang berbentuk silinder dan simetris. Stator merupakan bagian yang diam sebagai rangka tempat kumparan stator yang terpasang. Stator merupakan bagian yang diam sebagai rangka tempat kumparan stator yang terpasang. Stator terdiri dari : inti stator, kumparan stator, dan alur stator. Motor induksi satu fasa dilengkapi dengan dua kumparan stator yang dipasang terpisah, yaitu kumparan utama (*main winding*) atau sering disebut dengan kumparan berputar dan kumparan bantu (*auxiliary winding*) atau sering disebut dengan kumparan start. Rotor merupakan bagian yang berputar. Bagian ini terdiri dari : inti rotor, kumparan rotor dan alur rotor. Pada umumnya ada dua jenis rotor yang sering digunakan pada motor induksi, yaitu rotor belitan (*wound rotor*) dan rotor sangkar tupai (*squirrel cage rotor*).



**Gambar 2.1** Konstruksi Umum Motor Induksi Satu Fase

### 2.1.1 Kecepatan Motor

Kecepatan putaran motor induksi ditentukan oleh kecepatan medan magnet putar (sinkron) dan slip akibat beban yang dihubungkan dengan poros motor. Kecepatan sinkron terutama ditentukan oleh frekuensi sumber tegangan listrik yang dapat dihitung menggunakan Persamaan 2.1

$$n_s = \frac{120f}{P} \quad (2.1)$$

dengan,

$n_s$  = kecepatan sinkron (rpm).

$f$  = frekuensi jala-jala (Hz).

$P$  = jumlah kutub motor.

Kecepatan putaran motor selalu lebih rendah daripada kecepatan sinkron. Perbedaan relatif antara putaran rotor dan kecepatan sinkron disebut slip (S) yang dapat dihitung menggunakan Persamaan 2.2. Nilai slip dipengaruhi oleh beban. Pada keadaan tanpa beban, slip motor induksi sangat rendah sehingga putaran rotor mendekati kecepatan sinkron. Jika motor dibebani maka slip akan bertambah sehingga putaran motor menjadi berkurang.



$$S = \frac{n_s - n_r}{n_s} \times 100\% \quad (2.2)$$

$$n_r = (1 - S)n_s \quad (2.3)$$

dengan,

$S$  = slip

$n_r$  = kecepatan putaran rotor (rpm)

### 2.1.2 Pengaturan Kecepatan Motor Induksi

Pengaturan kecepatan motor induksi satu phasa dapat dilakukan dengan:

#### 1. Variasi Frekuensi Stator

Pengaturan putaran motor induksi dapat dilakukan dengan mengatur besar frekuensi yang dimiliki oleh motor induksi tersebut. Frekuensi berbanding lurus dengan kecepatan motor yang akan dihasilkan. Frekuensi juga berkaitan dengan torka akibat adanya beban yang diperlihatkan pada Gambar 2.2

#### 2. Perubahan Jumlah Kutub

Kecepatan motor induksi dapat pula diatur dengan mengubah jumlah kutubnya. Dimana, kecepatan motor induksi akan lebih cepat dengan menggunakan kutub yang lebih sedikit dibandingkan dengan berkutub banyak.

#### 3. Mengatur Tegangan Terminal

Pengaturan putaran motor induksi juga dapat dilakukan dengan mengubah tegangan terminal. Semakin besar nilai tegangan pada motor, maka kecepatan motor akan semakin besar seperti pada Gambar 2.3. Semakin besar beban yang diberikan pada motor mengakibatkan tegangan pada motor akan semakin besar mengikuti batas kecepatan motor. Tegangan pada motor induksi berpengaruh karena perubahan frekuensi mengakibatkan perubahan tegangan yang mengikuti. Tegangan tersebut dapat dihitung menggunakan Persamaan 2.4

$$E = 4,44 \cdot f \cdot n \cdot \phi \quad (2.4)$$

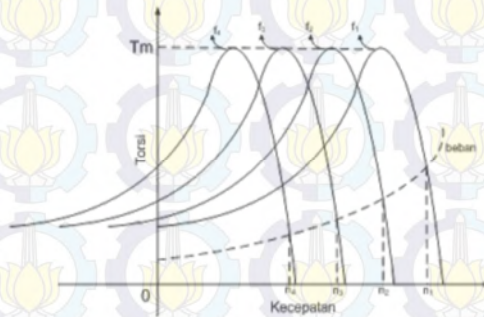
dimana

$E$  : tegangan induksi (V)

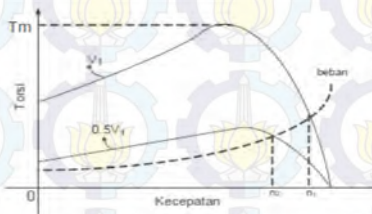
$f$  : frekuensi rotor saat berputar (Hz)

$n$  : jumlah belitan-belitan rotor

$\phi$  : fluks maksimum (Wb)



**Gambar 2.2** Karakteristik Pengaturan Kecepatan Terhadap Perubahan Frekuensi

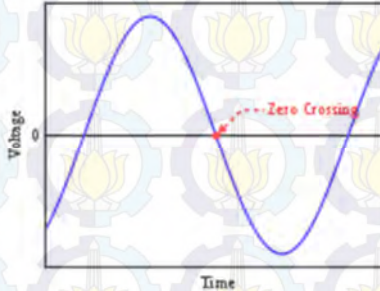


**Gambar 2.3** Karakteristik Perubahan Tegangan Terhadap Kecepatan Motor Induksi

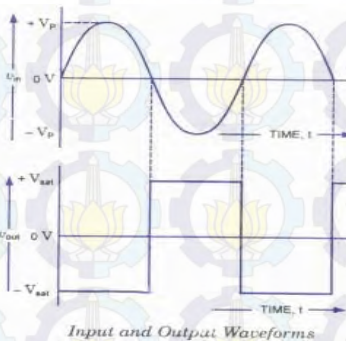
## 2.2 Zero Crossing Detector [2]

*Zero crossing detector* adalah rangkaian yang digunakan untuk mendeteksi perpotongan gelombang sinus AC 220 volt saat melewati titik tegangan nol seperti yang ditunjukkan pada Gambar 2.4. Prinsip kerja *Zero Crossing* dengan membandingkan tegangan AC terhadap tegangan referensi yang dihubungkan ke *ground* (0 volt). Pada Gambar 2.5 terlihat tegangan keluarannya adalah gelombang kotak dengan frekuensi sesuai dengan frekuensi AC-nya yaitu 50 Hz. Pada saat *Positive Going Transition* (PGT) atau *Negative Going Transition* (NGT)

inilah saat terjadi *zero*. PGT atau NGT inilah yang dibaca oleh mikrokontroler sebagai *zero*.



**Gambar 2.4** Grafik Tegangan *Input* dan Persilangan Nol



**Gambar 2.5** Gelombang *Input* dan *Output* Zero Crossing

### 2.3 TRIAC [2]

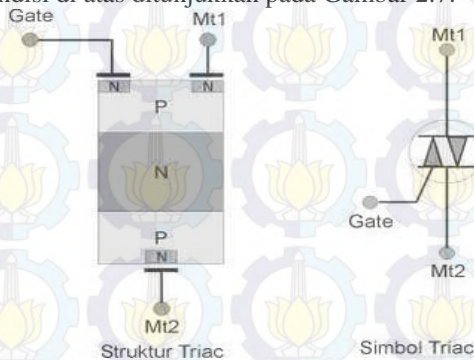
TRIAC, atau *Triode for Alternating Current* (Trioda untuk arus bolak-balik) adalah sebuah komponen elektronik yang ekuivalen dengan dua SCR yang disambungkan antiparalel dan kaki gerbangnya disambungkan bersama. Nama resmi untuk TRIAC adalah *Bidirectional Triode Thyristor*. Ini menunjukkan saklar dwi arah yang dapat mengalirkan arus listrik ke kedua arah ketika dipicu (dihidupkan). Ini dapat disulut baik dengan tegangan positif ataupun negatif pada elektroda gerbang. Sekali disulut, komponen ini akan terus menghantar hingga arus yang mengalir lebih rendah dari arus genggamnya, misal pada akhir paruh siklus dari arus bolak-balik.



### 2.3.1. Prinsip Kerja TRIAC

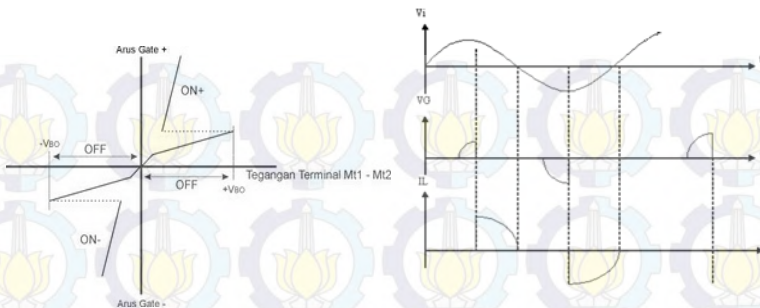
TRIAC merupakan komponen 3 elektroda: MT1, MT2, dan *gate* seperti yang terlihat pada Gambar 2.6. TRIAC biasanya digunakan pada rangkaian pengendali, penyaklaran, dan rangkian pemicu/*trigger*. Oleh karena aplikasi TRIAC yang demikian luas maka komponen TRIAC biasanya mempunyai dimensi yang besar dan mampu diaplikasikan pada tegangan 100V sampai 800V dengan arus beban dari 0.5A sampai 40A.

Jika terminal MT1 dan MT2 diberi tegangan jala-jala PLN dan *gate* dalam kondisi mengambang maka tidak ada arus yang dilewatkan oleh TRIAC (kondisi ideal) sampai pada tegangan '*break over*' TRIAC tercapai. Kondisi ini dinamakan kondisi *off* TRIAC. Apabila *gate* diberi arus positif atau negatif maka tegangan '*break over*' ini akan turun. Semakin besar nilai arus yang masuk ke *gate* maka semakin rendah pula tegangan '*break over*'nya. Kondisi ini dinamakan sebagai kondisi aktif TRIAC. Apabila TRIAC sudah aktif maka TRIAC akan dalam kondisi aktif selama tegangan pada MT1 dan MT2 di atas 0 volt. Apabila tegangan pada MT1 dan MT2 sudah mencapai 0 volt maka kondisi kerja TRIAC akan berubah dari *on* ke *off*. Apabila TRIAC sudah menjadi *off* kembali, TRIAC akan selamanya *off* sampai ada arus *trigger* ke *gate* dan tegangan MT1 dan MT2 melebihi tegangan '*break over*'nya. Kondisi di atas ditunjukkan pada Gambar 2.7.



Gambar 2.6 Struktur dan Simbol TRIAC





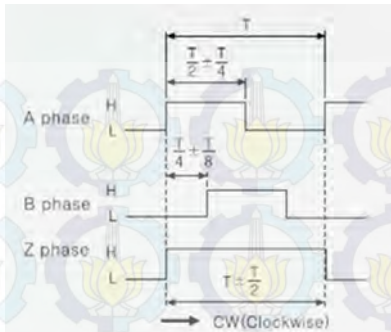
**Gambar 2.7** Daerah Kerja TRIAC

## 2.4 Incremental Rotary Encoder [3]

Instrumen *incremental rotary encoder* atau dikenal juga sebagai *incremental shaft encoder* adalah salah satu tipe dari peralatan *encoder* yang memberikan keluaran dalam format digital. Hal ini menyebabkan instrumen sejenis ini lebih nyaman digunakan pada aplikasi pengontrolan yang menggunakan komputer, sebagaimana pengukuran dibutuhkan dalam bentuk digital. Oleh karena itu, proses konversi dari sinyal analog ke digital tidak perlu lagi dilakukan.

Konsep dasar operasi instrumen *incremental rotary encoder* adalah instrumen ini mengukur nilai sesaat posisi angular dari sebuah *shaft* yang sedang berotasi dan menghasilkan pulsa-pulsa pada *channel-channel*-nya. Pulsa-pulsa yang dihasilkan ini berbentuk gelombang *square*.

Instrumen *incremental rotary encoder* biasanya memiliki tiga buah sinyal keluaran, yaitu sinyal A, sinyal B, dan sinyal Z, ditunjukkan dalam Gambar 2.8. Untuk sinyal A dan sinyal B, masing-masing sinyal keluaran tersebut saling *quadrature* yang berarti terjadi pergeseran fasa 90 derajat satu sama lain. Kedua sinyal tersebut selain memberikan nilai posisi *shaft* dari *encoder*, juga mampu menyediakan informasi mengenai arah putaran dari *shaft* misalnya berputar searah jarum jam atau berputar berlawanan arah jarum jam. Hal penting yang perlu diperhatikan hubungan antara sinyal A dan sinyal B adalah bahwa pergeseran fasa satu sama lain antara kedua sinyal tersebut harus berada dalam batas toleransi yang dapat diterima biasanya tidak melebihi 90 derajat sehingga proses perhitungan dapat berlangsung dengan akurat.



**Gambar 2.8** Tiga Sinyal Keluaran *Encoder*

## 2.5 DAC 0808 [6]

DAC0808 merupakan salah satu contoh konverter D/A (Digital to Analog). IC D/A ini menerapkan metode tangga R2R 8 bit yang dilengkapi dengan sumber arus acuan dan delapan buah transistor saklar untuk mengarahkan arus biner. Suatu tegangan dan hambatan eksternal dipergunakan untuk mengatur arus acuan pada nilai yang lazim berlaku yaitu 2 mA. DAC0808 mempunyai waktu pemantapan 150ns pada sumber tegangan  $\pm 5V$  dan resolusi  $\pm 1LSB$ . Konfigurasi pin ditunjukkan seperti pada Gambar 2.9 dengan penjelasan sebagai berikut:

- Pin 1 tidak dipakai (NC singkatan dari *no connection*).
- Pin 2 adalah penghubung ke *ground*.
- Pin 3 (VEE) harus dipasang pada -12V.
- Pin 4 adalah saluran balik dari *ground*, yang sifatnya sebagai *output* pembalik.
- Pin 5 s/d 12 merupakan 8 bit masukan data.
- Pin 13 (VCC) harus dipasang pada catu daya +5V.
- Pin 14 dihubungkan dengan catu daya positif melalui hambatan R14
- Pin 15 dihubungkan ke *ground* melalui hambatan R15.
- Kapasitor antar pin 16 dan pin 13 berfungsi untuk memberi kompensasi frekuensi bagi IC ini.

Keluaran DAC 0808 berupa besaran arus yang dapat dihitung menggunakan Persamaan 2.5

$$I_0 = K \left[ \frac{A_{N1}}{2} + \frac{A_{N2}}{4} + \frac{A_{N3}}{8} + \frac{A_{N4}}{16} + \frac{A_{N5}}{32} + \frac{A_{N6}}{64} + \frac{A_{N7}}{128} + \frac{A_{N8}}{256} \right] \quad (2.5)$$

dengan,

$$K \cong \frac{V_{REF}}{R14}$$

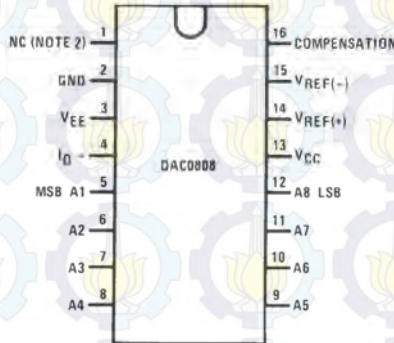
$A_N = 1$  saat biner bernilai *high*

$A_N = 0$  saat biner bernilai *low*

untuk mendapatkan nilai tegangan *output* maka nilai arus tersebut diubah menjadi besaran tegangan menggunakan Persamaan 2.6

$$V_0 = R_f \cdot I_0 \quad (2.6)$$

$$V_0 = R_f \cdot K \left[ \frac{A_{N1}}{2} + \frac{A_{N2}}{4} + \frac{A_{N3}}{8} + \frac{A_{N4}}{16} + \frac{A_{N5}}{32} + \frac{A_{N6}}{64} + \frac{A_{N7}}{128} + \frac{A_{N8}}{256} \right] \quad (2.7)$$



**Gambar 2.9** Konfigurasi Pin IC DAC 0808

## 2.6 Arduino UNO [4]

Arduino UNO adalah arduino *board* yang menggunakan mikrokontroler ATmega328 yang bersifat *open source*. Sifat *open source* arduino mempunyai keuntungan yaitu komponen yang digunakan tidak hanya bergantung pada satu merek, namun memungkinkan dapat digunakan oleh semua komponen yang ada dipasaran. Arduino UNO memiliki 14 pin digital (6 pin dapat digunakan sebagai *output* PWM), 6 *input* analog, sebuah 16 MHz osilator kristal, sebuah koneksi USB, sebuah konektor sumber tegangan, sebuah *header* ICSP, dan sebuah tombol *reset*. Bahasa pemrograman arduino merupakan bahasa C yang sudah disederhanakan syntax bahasa pemrogramannya.



Arduino UNO memiliki spesifikasi seperti pada Tabel 2.1 yang dapat beroperasi pada tegangan 6 sampai 20 volt. Jika arduino UNO diberi tegangan di bawah 7 volt, maka pin 5V akan menyediakan tegangan di bawah 5 volt dan arduino UNO mungkin bekerja tidak stabil. Jika diberikan tegangan melebihi 12 volt, Regulator tegangan kemungkinan menjadi terlalu panas dan merusak arduino UNO. Tegangan rekomendasi yang diberikan ke arduino UNO berkisar antara 7 sampai 12 volt.

Tabel 2.1 Tabel Spesifikasi Arduino UNO

Spesifikasi	Keterangan
Mikrokontroler	Atmega328
Tegangan Operasi	5 Volt
Tegangan <i>Input</i> (Rekomendasi)	7~12 Volt
Tegangan <i>Input</i> (batas)	5~20 Volt
Pin Digital I/O	14
Pin Analog <i>Input</i>	16
Arus DC per pin I/O	40 mA
Arus DC untuk pin 3.3V	150 mA
Memori <i>Flash</i>	32KB (0.5KB untuk <i>bootloader</i> )
SRAM	2 KB
EEPROM	1 KB
Kecepatan perwaktuan	16 Mhz

Masing-masing dari 14 pin digital arduino UNO dapat digunakan sebagai masukan atau keluaran menggunakan fungsi *pinMode()*, *digitalWrite()*, dan *digitalRead()*. Setiap pin beroperasi pada tegangan 5 volt. Setiap pin mampu menerima atau menghasilkan arus maksimum sebesar 40 mA dan memiliki resistor *pull-up* internal (diputus secara default) sebesar 20-30 KOhm. Sebagai tambahan, beberapa pin masukan digital memiliki kegunaan khusus sebagai berikut:

- Komunikasi serial: pin 0 (RX) dan pin 1 (TX), digunakan untuk menerima(RX) dan mengirim(TX) data secara serial.
- Interupsi Eksternal: pin 2 dan pin 3, pin ini dapat dikonfigurasi untuk memicu sebuah interupsi pada nilai rendah, sisi naik atau turun, atau pada saat terjadi perubahan nilai.
- *Pulse-width modulation* (PWM): pin 3,5,6,9,10 dan 11, menyediakan keluaran PWM 8-bit dengan menggunakan fungsi *analogWrite()*.



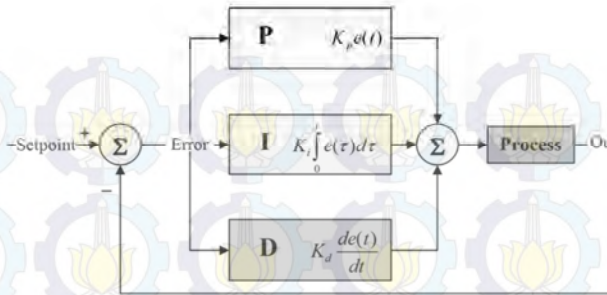
- *Serial Peripheral Interface (SPI)*: pin 10 (SS), 11 (MOSI), 12 (MISO) dan 13 (SCK), pin ini mendukung komunikasi SPI dengan menggunakan *SPI library*.
- LED: pin 13, terdapat *built-in* LED yang terhubung ke pin digital 13. Ketika pin bernilai *HIGH* maka LED menyala, sebaliknya ketika pin bernilai *LOW* maka LED akan padam.

Arduino UNO memiliki 6 masukan analog yang diberi label A0 sampai A5, setiap pin menyediakan resolusi sebanyak 10 bit (1024 nilai yang berbeda). Secara *default* pin mengukur nilai tegangan dari *ground* (0V) hingga 5V, walaupun begitu dimungkinkan untuk mengganti nilai batas atas dengan menggunakan pin AREF dan fungsi *analogReference()*. Sebagai tambahan beberapa pin masukan analog memiliki fungsi khusus yaitu pin A4 (SDA) dan pin A5 (SCL) yang digunakan untuk komunikasi *Two Wire Interface (TWI)* atau *Inter Integrated Circuit (I2C)* dengan menggunakan *Wire library* (Arduino).

## 2.7 Kontrol PID [7][8]

Proporsional-integral-derivatif (PID) sering disebut sebagai kontroler 'jangka tiga'. Saat ini salah satu pengendali yang paling sering digunakan dalam industri proses. Dalam kontroler PID, variabel kontrol yang dihasilkan dari jangka waktu proporsional untuk *error*, jangka waktu yang mana merupakan integral dari *error*, dan jangka waktu yang mana merupakan turunan dari *error*.

Salah satu tugas pengendali PID adalah mereduksi sinyal kesalahan, yaitu perbedaan antara sinyal setting dan sinyal aktual. Semakin cepat respon pengendali dan semakin kecil nilai kesalahan yang didapat maka semakin baik pula pengendali PID tersebut. Blok diagram pengendalai PID diperlihatkan pada Gambar 2.10 dimana setiap persamaan dari P, I, D saling berhubungan untuk mendapatkan *output* yang diinginkan.



**Gambar 2.10** Blok Diagram Kontrol PID

### 2.7.1. Kontrol Proporsional

Kontrol proporsional berfungsi untuk memperkuat sinyal kesalahan penggerak (sinyal *error*), sehingga akan mempercepat keluaran sistem mencapai titik referensi. Hubungan antara *input* kontroler  $u(t)$  dengan sinyal  $e(t)$  terlihat pada Persamaan 2.8

$$u(t) = K_p e(t) \quad (2.8)$$

*Error* dikalikan dengan *gain*  $K_p$ . Sebuah *gain* yang sangat tinggi dapat menyebabkan ketidakstabilan, dan *gain* yang sangat rendah dapat menyebabkan sistem rusak.

### 2.7.2. Kontrol Integral

Kontrol integral pada prinsipnya bertujuan untuk menghilangkan kesalahan keadaan tunak (*offset*) yang biasanya dihasilkan oleh kontrol proporsional. Hubungan antara *output* kontrol integral  $u(t)$  dengan sinyal  $e(t)$  terlihat pada Persamaan 2.9

$$u(t) = K_i \int_u^t e(t) dt \quad (2.9)$$

Integral dari *error* diambil dan dikalikan dengan *gain*  $K_i$ . *Gain* dapat disesuaikan untuk mengarahkan nilai *error* ke nol selama waktu yang dibutuhkan. Sebuah *gain* terlalu tinggi dapat menyebabkan osilasi dan *gain* terlalu rendah dapat menyebabkan respon lamban.

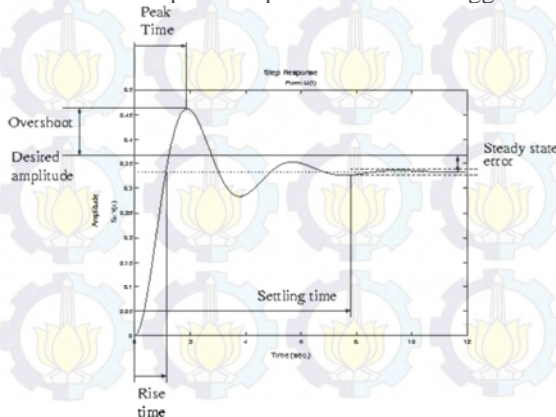
### 2.7.3. Kontrol Derivatif

Kontrol derivatif dapat disebut pengendali laju, karena *output* kontroler sebanding dengan laju perubahan sinyal *error*. Hubungan antara *output* derivatif  $u(t)$  dengan sinyal *error*  $e(t)$  terlihat pada Persamaan 2.10

$$u(t) = K_d \frac{de(t)}{dt} \quad (2.10)$$

Turunan dari *error* dikalikan dengan *gain*  $K_d$ . Sekali lagi, jika *gain* terlalu tinggi sistem dapat berosilasi dan jika *gain* terlalu rendah respon mungkin lamban. Maka, gabungan dari nilai  $K_p$ ,  $K_i$ ,  $K_d$  akan menghasilkan gelombang seperti pada Gambar 2.11 dan keterangan dari setiap gelombang seperti berikut:

1. *Time Constant* ( $t$ ) : Ukuran waktu yang menyatakan kecepatan respon
2. *Settling Time* (TS): Ukuran waktu yang menyatakan respon telah masuk  $\pm 5\%$  atau  $\pm 2\%$  atau  $\pm 0,5\%$  dari respon *steady state*.
3. *Rise Time* (TR) : Ukuran waktu yang menyatakan keberadaan suatu respon, yang di ukur mulai respon 5% s/d 95% dari respon *steady state* (dapat pula 10% s/d 90%).
4. *Delay Time* (TD) : Ukuran waktu yang menyatakan faktor keterlambatan respon output terhadap input, di ukur mulai  $t = 0$  s/d respon mencapai 50% dari respon *steady state*.
5. *Overshoot* : seberapa besar peak level lebih tinggi dari *steady state*



**Gambar 2.11**Waktu Respon Sistem

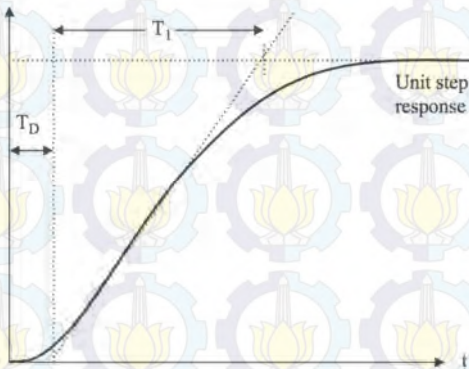


#### 2.7.4. Metode Ziegler-Nichols

Metode Ziegler-Nichols dan *good gain method* merupakan salah satu metode selain dari perhitungan matematika yang digunakan untuk mendapatkan nilai parameter PID yang sesuai untuk sistem. Metode ini melakukan percobaan pendekatan secara pengujian untuk mendapatkan parameter PID. Terdapat 2 metode Ziegler Nichols ini yang digunakan pada sistem *loop* terbuka

Untuk *loop* terbuka, langkah pertama menemukan parameter *plant* dengan menerapkan langkah untuk sistem *loop* terbuka. Parameter *plant*  $K$ ,  $T_d$ , dan  $T_i$  kemudian akan dijelaskan pada Gambar 2.12

Kemudian Ziegler dan Nichols menyarankan menggunakan pengaturan kontroler PID yang diberikan dalam Gambar 2.13 saat *loop* tertutup. Parameter ini didasarkan pada konsep meminimalkan integral dari kesalahan mutlak setelah menerapkan langkah perubahan ke *setpoint*



Gambar 2.12 Grafik Parameter *Plant*  $K$ ,  $T_d$ , dan  $T_i$

Controller	$K_p$	$T_i$	$T_d$
Proportional	$\frac{T_1}{KT_D}$		
PI	$\frac{0.9T_1}{KT_D}$	$3.3T_D$	
PID	$\frac{1.2T_1}{KT_D}$	$2T_D$	$0.5T_D$

Gambar 2.13 Pengaturan Sistem *Loop* Terbuka Ziegler-Nichols



### 2.7.5 Persamaan PID

Gabungan dari proposional, integral, dan derivatif dapat dinyatakan pada Gambar 2.14 yang menunjukkan diagram blok klasik waktu kontinu kontroler PID.

Menyetel kontroler meliputi penyesuaian parameter  $K_p$ ,  $K_d$  dan  $K_i$  untuk mendapatkan respon yang memuaskan. Karakteristik kontroler PID diketahui dengan baik dan benar, dan sebagian besar pengendali modern berdasarkan pada beberapa bentuk PID. Hubungan *input-output* kontroler PID dapat dinyatakan sebagai Persamaan 2.11

$$u(t) = K_p \left[ e(t) + \frac{1}{T_i} \int e(t) dt + T_d \frac{de(t)}{dt} \right] \quad (2.11)$$

dengan,

$u(t)$  = sinyal *output* dari kontroler

$K_p$  = Konstanta Proposional

$T_i$  = waktu integral

$T_d$  = waktu derivatif

$e(t)$  = sinyal *error*

Perhatikan bahwa Persamaan 2.11 kadang-kadang ditulis sebagai

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} + u_0, \quad (2.12)$$

dimana,

$$K_i = \frac{K_p}{T_i} \text{ dan } K_d = K_p T_d. \quad (2.13)$$

Setelah mendapatkan persamaan PID di atas maka untuk memasukkan parameter PID ke dalam program menggunakan persamaan PID kontrol seperti Gambar 2.15. Sehingga, didapatkan parameter sebagai berikut:

$$\alpha = K_p T / T_i \quad (2.14)$$

$$\beta = K_p T_D / T \quad (2.15)$$

Dengan mengacu pada Gambar 2.15 dapat ditulis persamaan perbedaan berikut:

untuk bagian proporsional,

$$w_k = K_p e_k \quad (2.16)$$

untuk bagian integral,

$$p_k = \alpha e_k + p_{k-1} \quad (2.17)$$

dan untuk bagian derviatif,

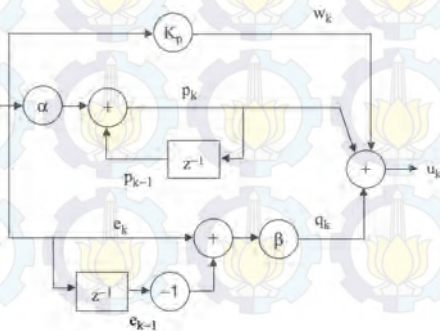
$$q_k = \beta(e_k - e_{k-1}) \quad (2.18)$$

Maka, keluaran yang dihasilkan sebagai Persamaan 2.19

$$u_k = w_k + p_k + q_k \quad (2.19)$$



**Gambar 2.14** Continuous-time Sistem Kontrol PID



**Gambar 2.15**Kontrol PID Dalam Struktur Paralel

### 2.7.6 Metode *Good Gain*

Metode *good gain* merupakan salah satu cara untuk mendapatkan parameter PID yang sesuai dengan sistem. Metode ini berbeda dengan metode Ziegler Nichols dimana pada metode ini tidak mengatur sistem kontrol sampai pada keadaan berosilasi. Adapun prosedur yang harus dilakukan dalam metode ini untuk menentukan parameter PID yaitu sebagai berikut:

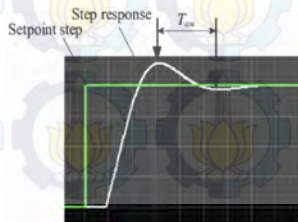
1. Proses pertama dilakukan pada sistem *loop* tertutup dimana sinyal kontrol dimasukkan secara manual.
2. Kontrol proporsional dimasukan secara bertahap dimulai dari 0 atau 1 ( $K_p = 0$ ), dan untuk waktu integral dan derivatif diset  $\infty$  dan 0 ( set  $T_i = \infty$  dan  $T_d = 0$ ). Penambahan atau pengurangan nilai  $K_p$  berpengaruh terhadap respon *overshoot* suatu sistem.
3. Set waktu integral  $T_i$  sesuai Persamaan 2.20

$$T_i = 1,5T_{ou} \quad (2.20)$$

Dimana pada Gambar 2.16 menunjukkan  $T_{ou}$  adalah waktu respon suatu sistem setelah mengalami *overshoot* menuju *steady state*.

4. Cek kestabilan sistem kontrol setelah digunakan langkah *setpoint* karena ini merupakan permulaan untuk parameter integral. Parameter integral sangat mempengaruhi parameter proporsional. Jika pada kontrol PI kestabilan sistem menjadi buruk maka langkah yang harus dilakukan adalah mengurangi  $K_p$  80% dari harga aslinya.
5. Jika ingin memasukkan parameter derivatif agar kontroler menjadi PID kontrol, maka *setting* sesuai dengan Persamaan 2.21

$$T_d = \frac{T_i}{4} \quad (2.21)$$



**Gambar 2.16** Respon  $T_{ou}$



## 2.8 LabVIEW [5]

LabVIEW (*Laboratory Virtual Instrument Engineering Workbench*) adalah bahasa pemrograman grafis yang menggunakan ikon bukan baris teks untuk membuat aplikasi. *Software* ini digunakan khusus untuk pemrosesan dan visualisasi data dalam bidang akuisisi data, kendali dan instrumentasi, serta otomatisasi industri. Berbeda dengan bahasa pemrograman berbasis teks, di mana petunjuk menentukan eksekusi program. LabVIEW menggunakan data *flow* pemrograman, di mana aliran data menentukan eksekusi.

LabVIEW adalah berorientasi objek grafis bahasa komputer yang dikembangkan untuk memfasilitasi perangkat keras dan perangkat lunak. Komunikasi LabVIEW adalah bahasa komputer yang lengkap yang dapat digunakan seperti *Basic*, FORTRAN, atau C. Dalam LabVIEW kita bisa untuk menciptakan instrumentasi *virtual* yang estetik tampak seperti instrumentasi nyata tetapi dikendalikan oleh program komputer yang canggih. Ada beberapa tingkatan instrumentasi *virtual* akuisisi data yang membuatnya mudah untuk mengontrol aliran data, dan pemrosesan sinyal banyak dan algoritma analisis datang dengan perangkat lunak sebagai instrumentasi *virtual premade*. Di dalam kelas, kesamaan antara instrumen *virtual* dan nyata membantu mahasiswa memahami bagaimana informasi dilewatkan antara komputer dan instrumen terpasang.

Program pada LabVIEW yang disebut sebagai VI (*Virtual Instrument*). Setiap VI menggunakan fungsi-fungsi yang menggerakkan masukan dari pemakai antarmuka atau sumber lain dan menampilkan informasi itu atau memindahkannya ke file lain atau ke komputer lain. VI menyediakan tiga komponen yaitu :

- Panel muka (*front panel*), yang melayani antarmuka pengguna. *Front panel* merupakan *interface* antar pengguna (*user*) dengan program. Di dalam *front panel* terdapat kontrol (*input*) dan indikator (*output*) yang dikenal dengan Kontrol Palet. Kontrol pada *front panel* dapat berupa knop, tombol, *dial*, dan lainnya. Sedangkan untuk indikator (*output*) dapat berupa LED, grafik, dan tampilan lainnya. Kontrol menirukan *input* instrumen dan menyuplai data ke diagram blok pada VI yang bersangkutan. Indikator menirukan instrumen keluaran dan menampilkan data. Untuk menampilkan Kontrol Palet maka pada *front panel* dapat diaktifkan View >> *Control Palette*.
- Diagram blok, berisi kode sumber grafis yang menggambarkan



fungsi-fungsi VI. Setelah membangun *front panel*, perlu menambahkan kode menggunakan grafis yang mewakili fungsi untuk mengendalikan objek - objek *front panel*. Blok diagram berisi *source code* grafis ini. Objek - objek *front panel* tampak seperti terminal pada diagram blok. *Virtual Instrument* menunjukkan beberapa objek diagram blok seperti terminal - terminal, fungsi - fungsi dan alur. Pada diagram blok terdapat *Function Pallette*, yang menyediakan *visible items* untuk membentuk suatu program. Jika tidak ditampilkan pada diagram blok *windows* maka dapat diaktifkan melalui *Window >> Show Functions Pallette*, atau klik kanan area pada diagram *window*.



### **BAB III**

#### **PERANCANGAN DAN PEMBUATAN ALAT**

Perancangan alat pada bab ini terdiri dari perancangan perangkat keras (*hardware*) dan perancangan perangkat lunak (*software*).

Perancangan perangkat keras (*hardware*) dibagi menjadi 2 bagian, yaitu meliputi perancangan elektronik dan juga perancangan mekanik. Perancangan elektronik terdiri dari skematik I/O Arduino UNO, pengendalian motor menggunakan *driver* TRIAC, perancangan ZCD (*Zero Crossing Detector*), perancangan DAC 8 bit, serta perancangan sensor *rotary encoder*. Sedangkan untuk perancangan mekanik, meliputi perancangan kopel sensor dan perancangan rem mekanik.

Untuk perancangan perangkat lunak (*software*) meliputi perancangan pemrograman Arduino UNO untuk *Interface* dan PID, Arduino UNO untuk *switching* TRIAC, dan perancangan pemrograman LabVIEW.

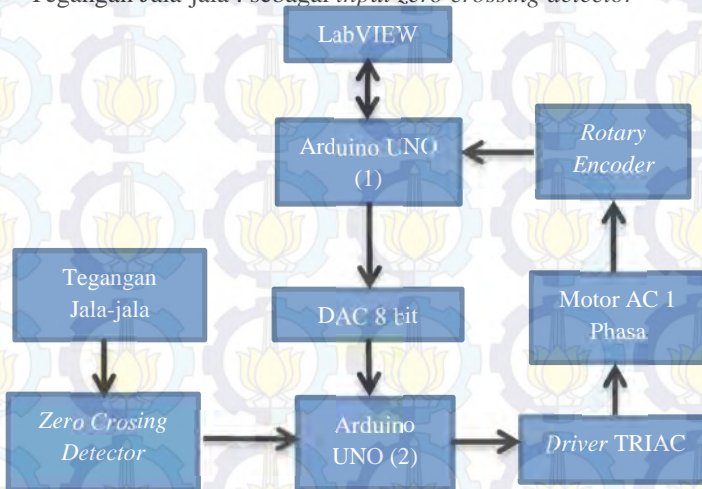
#### **3.1. Blok Fungsional Sistem**

Pada bab ini dibahas mengenai perancangan alat secara keseluruhan. Dijelaskan pula lebih terinci tiap-tiap bagian atau blok-blok penyusun alat ini pada Gambar 3.1 yang berupa blok fungsional sistem.

Tahapan-tahapan yang ada pada blok fungsional untuk perancangan Tugas Akhir ini sebagai berikut :

1. LabVIEW : digunakan untuk mengirim parameter-parameter pengaturan seperti *setpoint*, kp, ki, dan kd. Selain itu juga menerima data berupa nilai kecepatan motor
2. Arduino UNO (1) : digunakan untuk menerima parameter yang dikirim dari LabVIEW, menerima sinyal pulsa dari *rotary encoder* untuk diubah menjadi nilai kecepatan, mengirim data nilai kecepatan ke LabVIEW, dan mengirim *output* kecepatan berupa data digital ke DAC 8 bit.
3. DAC 8 bit : digunakan untuk mengubah data digital yang dikirim oleh Arduino UNO (1) menjadi data analog untuk dikirim ke Arduino UNO (2)
4. *Zero Crossing Detector* : digunakan untuk mendeteksi persilangan nol tegangan jala-jala PLN
5. Arduino UNO (2) : digunakan untuk menerima data analog DAC dan melakukan *switching driver*

6. *Driver TRIAC* : digunakan untuk mengatur kecepatan motor dengan metode *switching*
7. *Rotary encoder* : digunakan untuk membaca kecepatan motor dan mengeluarkannya dalam bentuk sinyal pulsa
8. Motor AC 1 phasa : *plant* yang akan dikontrol
9. Tegangan Jala-jala : sebagai *input zero crossing detector*



**Gambar 3.1** Blok Fungsional Sistem

### 3.2. Perancangan Perangkat Keras (*Hardware*)

Perancangan perangkat keras (*hardware*) dibagi menjadi 2 bagian, yaitu meliputi perancangan elektronik dan juga perancangan mekanik. Perancangan elektronik terdiri dari perancangan *power supply*, skematik I/O Arduino UNO, perancangan DAC 8 bit, perancangan ZCD (*Zero Crossing Detector*), pengendalian motor menggunakan *driver TRIAC*, serta perancangan sensor *rotary encoder*. Selanjutnya pada perancangan mekanik, meliputi perancangan kopel sensor dan perancangan rem mekanik.

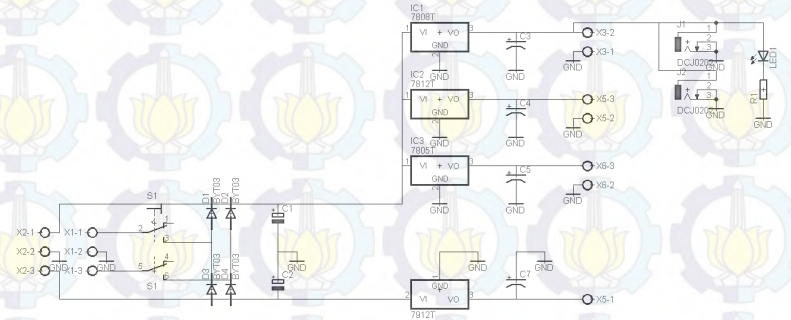
#### 3.2.1. Perancangan Elektronik

##### 3.2.1.1. Perancangan *Power Supply*

*Power supply* digunakan untuk memberikan suplai tegangan sesuai yang dibutuhkan oleh rangkaian. Tegangan jala-jala PLN



diturunkan menggunakan *transformator step down* dan diubah menjadi tegangan DC menggunakan penyearah gelombang penuh. Tegangan DC tersebut difilter menggunakan kapasitor sebelum dan sesudah diregulasi menggunakan IC *regulator* LM78XX dan LM79XX untuk mengurangi riak. Berikut Gambar 3.2 rangkaian *power supply* secara keseluruhan :



**Gambar 3.2 Rangkaian Power Supply**

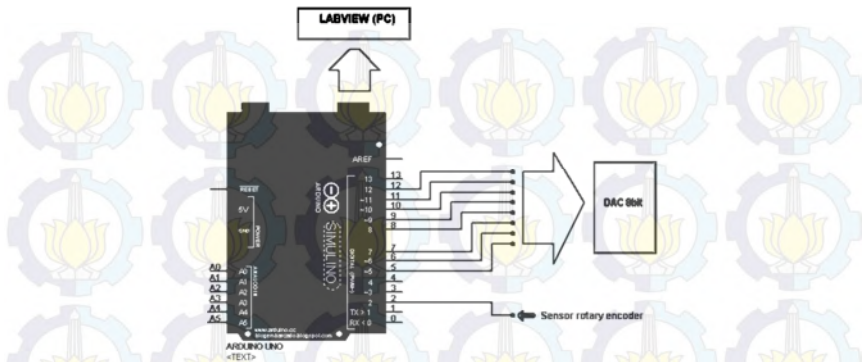
### 3.2.1.2. Skematik Arduino UNO (1)

Pada Gambar 3.3 Skematik Arduino UNO dapat diketahui mengenai penyambungan / pengkoneksian *port - port* yang digunakan pada *rotary encoder* dan DAC 8 bit.

Pin - pin pada Arduino UNO (1) yang digunakan pada Tugas Akhir ini dijelaskan pada Tabel 3.1 sebagai berikut :

**Tabel 3.1 Pin Arduino UNO (1) yang Digunakan**

PIN	FUNGSI
2	Membaca sinyal pulsa <i>encoder</i>
5	Digital <i>Output</i> bit ke-1
6	Digital <i>Output</i> bit ke-2
7	Digital <i>Output</i> bit ke-3
8	Digital <i>Output</i> bit ke-4
9	Digital <i>Output</i> bit ke-5
10	Digital <i>Output</i> bit ke-6
11	Digital <i>Output</i> bit ke-7
12	Digital <i>Output</i> bit ke-8



**Gambar 3.3** Skematik Arduino UNO (1)

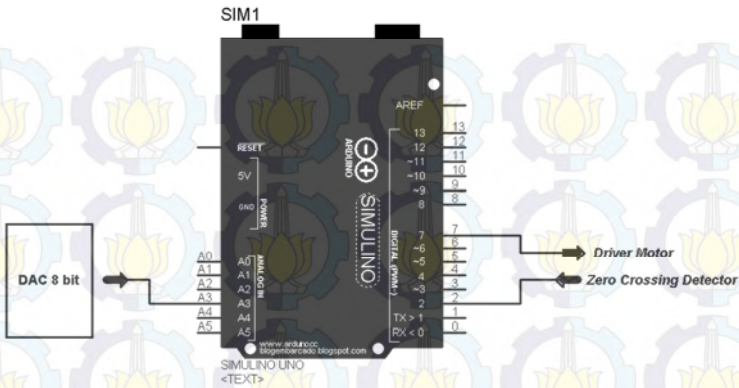
### 3.2.1.3. Skematik Arduino UNO (2)

Arduino UNO (2) digunakan untuk melakukan proses *switching driver* TRIAC AC 1 fasa seperti pada Gambar 3.4. Digunakannya Arduino UNO yang kedua ini dikarenakan dibutuhkan pembacaan *interrupt* dan pada Arduino UNO yang pertama telah terjadi proses pembacaan pin *interrupt*. Meskipun Arduino UNO memiliki 2 pin *Interrupt*, kedua proses tidak dapat dikerjakan pada satu Arduino karena memiliki waktu tunda yang berbeda.

Pin - pin pada Arduino UNO (2) yang digunakan pada Tugas Akhir ini dijelaskan pada Tabel 3.2 sebagai berikut :

**Tabel 3.2** Pin Arduino UNO (2) yang Digunakan

PIN	FUNGSI
A3	Membaca data analog dari DAC 8 bit
2	Membaca sinyal pulsa dari <i>Zero Crossing Detector</i>
7	<i>Switching Driver</i> TRIAC



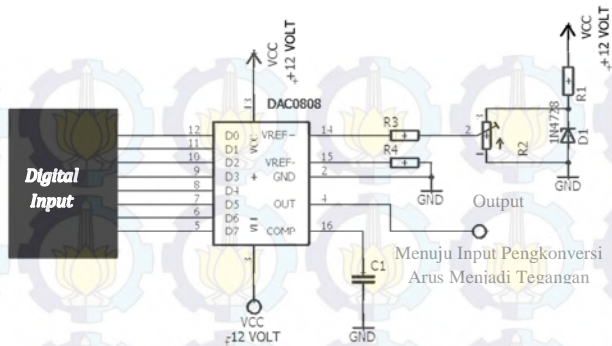
**Gambar 3.4** Skematik Arduino UNO (2)

#### 3.2.1.4. Perancangan DAC 8 bit

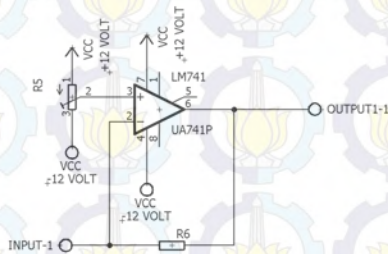
Digital to Analog *Converter* (DAC) diunakan untuk mengubah suatu data digital berupa nilai biner menjadi suatu tegangan analog. Pembuatan DAC dikarenakan Arduino UNO tidak mampu menghasilkan *output* berupa tegangan analog. Arduino UNO hanya mampu menghasilkan suatu sinyal digital 0 volt atau 5 volt. Sedangkan, pada Arduino UNO (2) yang akan digunakan untuk melakukan *switching driver* TRIAC membutuhkan tegangan analog sebagai lama waktu tunda *switching driver* TRIAC sehingga akan didapatkan nilai tegangan yang diinginkan. Maka, dibuat rangkaian suatu pengkonversi data digital menjadi tegangan analog menggunakan IC DAC0808 seperti pada Gambar 3.5

Namun DAC0808 ini memiliki keluaran berupa arus, maka arus tersebut harus dikonversikan terlebih dahulu menjadi tegangan menggunakan IC LM741 seperti Gambar 3.6 yang merupakan rangkaian konversi arus menjadi tegangan

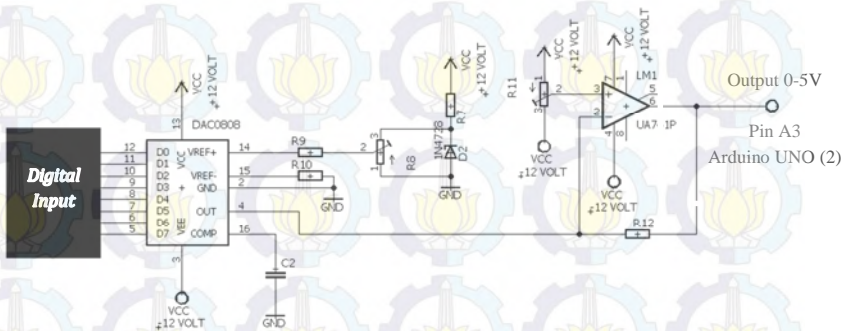
Rangkaian keseluruhan merupakan gabungan antara rangkaian IC DAC0808 dengan pengkonversi arus menjadi tegangan IC LM741. Sehingga didapatkan suatu rangkaian yang dapat mengubah suatu data digital (8 bit) menjadi suatu nilai tegangan analog seperti Gambar 3.7 yaitu keseluruhan rangkaian DAC 8bit



**Gambar 3.5** Skematik DAC0808



**Gambar 3.6** Pengkonversi Arus Menjadi Tegangan



**Gambar 3.7** Rangkaian Keseluruhan DAC 8bit



### 3.2.1.5. Perancangan Zero Crossing Detector

*Zero Crossing Detector* adalah rangkaian yang digunakan untuk mendeteksi gelombang sinus tegangan jala-jala AC 220 V ketika melalui persimpangan titik nol. Rangkaian *Zero Crossing Detector* pada Tugas Akhir ini menggunakan optocoupler IC 4N35 seperti Gambar 3.8 yaitu rangkaian *Zero Crossing Detector*

Pada rangkaian ZCD tersebut dibutuhkan sinyal AC yang disearahkan dengan menggunakan rangkaian penyearah gelombang penuh. Sinyal AC yang telah diubah menjadi sinyal DC masuk ke LED optocoupler IC 4N35 melalui sebuah resistor. Nilai resistor dapat diketahui dengan cara sebagai berikut :

$$V_{rms} = 25V$$

$$I_f = 10mA$$

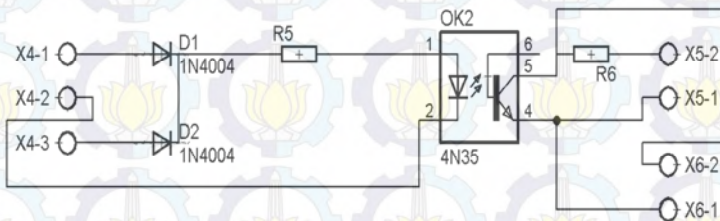
$$V_f = 1,3 V$$

$$V_m = \frac{V_{rms}}{\sqrt{2}} = \frac{25}{\sqrt{2}} = 35,35 V$$

$$V_{res} = 35 - V_f = 35 - 1,3 = 32,7 V$$

$$R = \frac{V_{res}}{I_f} = \frac{32,7}{10 \cdot 10^{-3}} = 3,270\Omega \rightarrow 3,3K\Omega$$

Kemudian tegangan 5VDC masuk ke bagian *receiver* optocoupler yang telah di seri dengan resistor *pull up*. Pin X5-2 pada Gambar 3.8 merupakan pin yang akan masuk ke pin *interrupt* Arduino UNO (2).

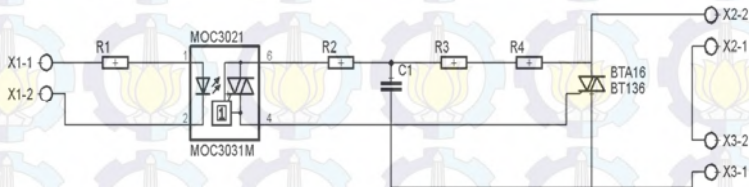


**Gambar 3.8** Rangkaian *Zero Crossing Detector*

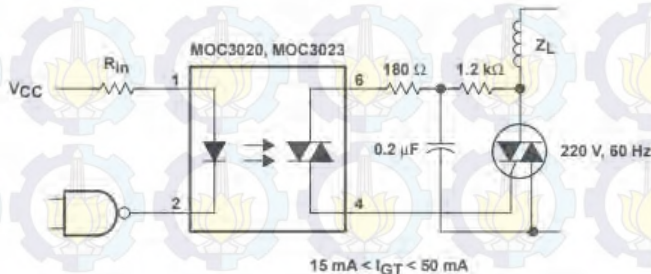
### 3.2.1.6. Perancangan *Driver TRIAC*

*Driver TRIAC* digunakan untuk mengatur tegangan yang masuk ke motor dengan metode *switching*. *Switching TRIAC* Arduino UNO (2) menggunakan tegangan rendah DC. Sedangkan tegangan yang masuk ke motor adalah tegangan jala-jala PLN 220-240VAC. Oleh karena itu pada rangkaian *driver* ini dibutuhkan suatu pemisah antara tegangan rangkaian kontrol yang berupa tegangan rendah DC terhadap rangkaian daya yang berupa tegangan tinggi AC seperti pada Gambar 3.9 yaitu rangkaian *driver TRIAC* AC 1 fasa

Pada rangkaian *driver* tersebut digunakan MOC3021 sebagai isolasi antara tegangan rendah DC dengan tegangan tinggi AC. TRIAC BTA16 digunakan sebagai saklar antara sumber tegangan dan beban. Rangkaian tersebut sesuai dengan rangkaian yang ada di dalam *datasheet* IC MOC3021 dengan beban induktif dengan arus *gate* antara 15mA-50mA seperti yang terlihat pada Gambar 3.10



Gambar 3.9 *Driver TRIAC* AC 1 Fasa



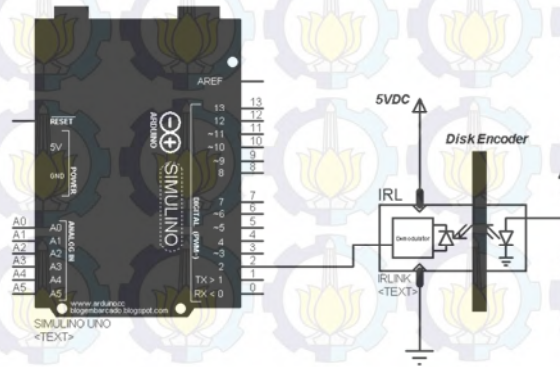
Gambar 3.10 Rangkaian MOC3021

### 3.2.1.7. Perancangan *Sensor Rotary Encoder*

*Sensor rotary encoder* digunakan untuk membaca kecepatan putaran motor secara aktual. Jenis *sensor rotary encoder* yang digunakan adalah *incremental rotary encoder* yang memiliki 25 lubang

pada bagian *shaft*-nya. Sehingga *rotary encoder* ini menghasilkan 25 pulsa tiap putaran (25ppr). Semakin besar nilai ppr dari suatu *encoder*, maka akan semakin teliti pembacaan kecepatannya.

Namun di sisi lain, karena sensor ini menggunakan fungsi *external interrupt* dari Arduino UNO seperti yang terlihat pada Gambar 3.11, maka harus dipertimbangkan besar ppr dengan kemampuan pembacaan data pada pin *interrupt* Arduino UNO



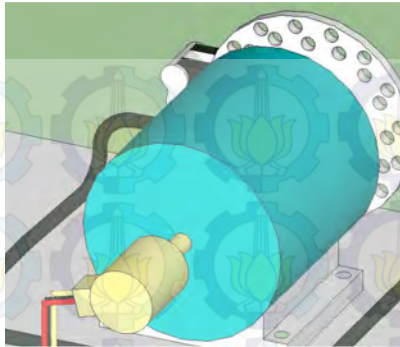
**Gambar 3.11** Sensor Rotary Encoder

### 3.2.2. Perancangan Mekanik

#### 3.2.2.1. Perancangan Kopel Sensor

Perancangan kopel sensor disini maksudnya adalah penghubung antara rotor motor dengan sensor *rotary encoder* seperti yang terlihat pada Gambar 3.12. Sensor *rotary encoder* dipasang dibagian depan motor sehingga dibuatkan sebuah kopel atau penghubung dibagian depan motor tersebut menggunakan kopel 1:1. Perancangan asli dapat dilihat pada lampiran A-2 hingga A-4

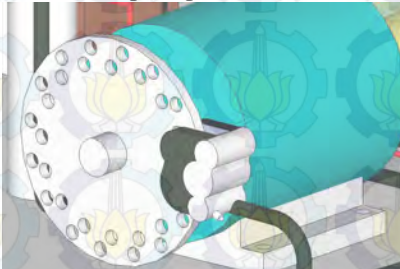




**Gambar 3.12** Kopel Sensor

#### **3.2.2.2. Perancangan Rem Mekanik**

Rem digunakan sebagai beban motor untuk mengetahui respon PID terhadap perubahan beban. Rem motor menggunakan sebuah piringan cakram yang dipasang di bagian depan motor. Proses pengereman dilakukan dengan cara mencengkram piringan dengan sebuah perangkat mekanik seperti pada Gambar 3.13



**Gambar 3.13** Perancangan Rem Mekanik

### **3.3. Perancangan Perangkat Lunak (Software)**

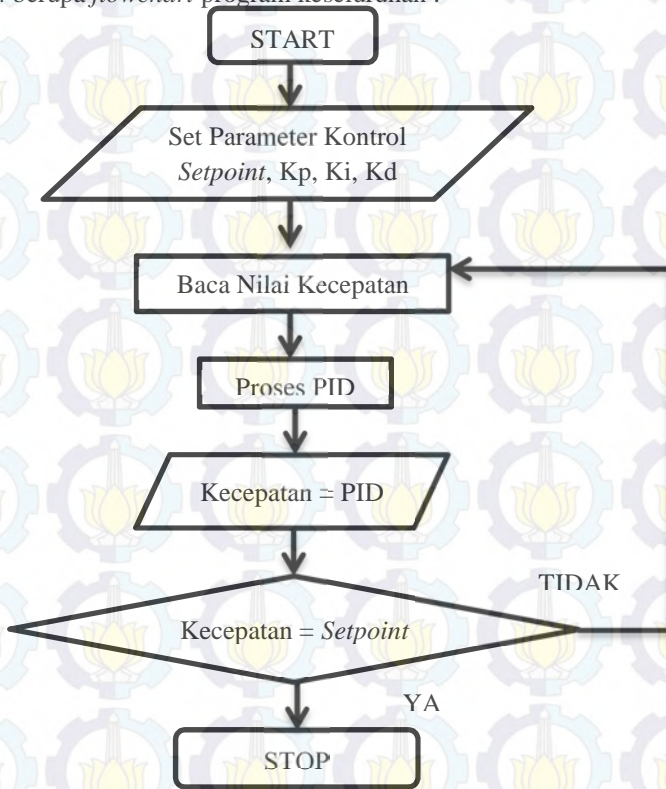
Perancangan perangkat lunak (*software*) pada Tugas Akhir ini terdiri dari pemrograman Arduino UNO dan pemrograman LabVIEW

#### **3.3.1. Pemrograman Arduino UNO**

Arduino adalah pengendali mikro yang bersifat *open-source*, diturunkan dari *wiring* platform, dirancang untuk memudahkan penggunaan elektronik dalam berbagai bidang. Perangkat kerasnya memiliki prosesor Atmel AVR dan perangkat lunaknya memiliki bahasa



pemrograman sendiri. Perangkat lunak yang digunakan dalam pemrograman Arduino UNO adalah Arduino IDE (*Integrated Development Environment*). Bahasa yang dipakai dalam Arduino bukan assembler yang relatif sulit, tetapi bahasa C yang disederhanakan dengan bantuan pustaka-pustaka (*libraries*) Arduino. Berikut Gambar 3.14 berupa *flowchart* program keseluruhan :



**Gambar 3.14** *Flowchart* Program

#### 3.3.1.1. Program Pembaca Kecepatan *Rotary Encoder*

Program Pembaca kecepatan dimaksudkan untuk mendapatkan nilai kecepatan motor. *Rotary encoder* yang menghasilkan sinyal pulsa

perlu diterjemahkan menjadi suatu nilai kecepatan dengan menggunakan rumus sebagai berikut :

$$n = \frac{60 \cdot f}{T \cdot ppr} \quad (3.1)$$

dengan,

$n$  = kecepatan (rpm)

$f$  = jumlah pulsa yang dihasilkan *rotary encoder*

$ppr$  = jumlah pulsa yang dihasilkan dalam satu putaran

$T$  = waktu *sampling* (detik)

Dalam pembacaan kecepatan digunakan fungsi *external interrupt* dan *interrupt timer*. Arduino UNO memiliki dua pin *interrupt*, yaitu INT0 dan INT1. INT0 berada pada pin 2 dan INT1 berada pada pin 3 Arduino UNO. Pada program kali ini dipergunakan pin *external interrupt* 2 (INT0). Sedangkan *interrupt timer* dipergunakan untuk memberikan *time sampling*. Semakin besar *time sampling* yang dipergunakan, akan semakin besar nilai *error* pembacaan kecepatannya. Dalam program ini dipergunakan *time sampling* sebesar 0.1 detik (100000  $\mu$ S). Berikut Gambar 3.15 berupa program pembacaan kecepatan motor :

```

sketch_may11a $
#include <TimerOne.h>
unsigned int counter = 0;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  Timer1.initialize(100000);
  attachInterrupt(0, docount, RISING);
  Timer1.attachInterrupt( timerIsr );
}

void docount()
{
  counter++;
}

void timerIsr()
{
  Timer1.detachInterrupt();
  rpm = ((counter * 60) / (25 * 0.1));
  Serial.println(rpm);
  counter=0;
  Timer1.attachInterrupt( timerIsr );
}

void loop() { }

```

**Gambar 3.15** Program Arduino Pembaca Kecepatan

### 3.3.1.2. Program Komunikasi Serial

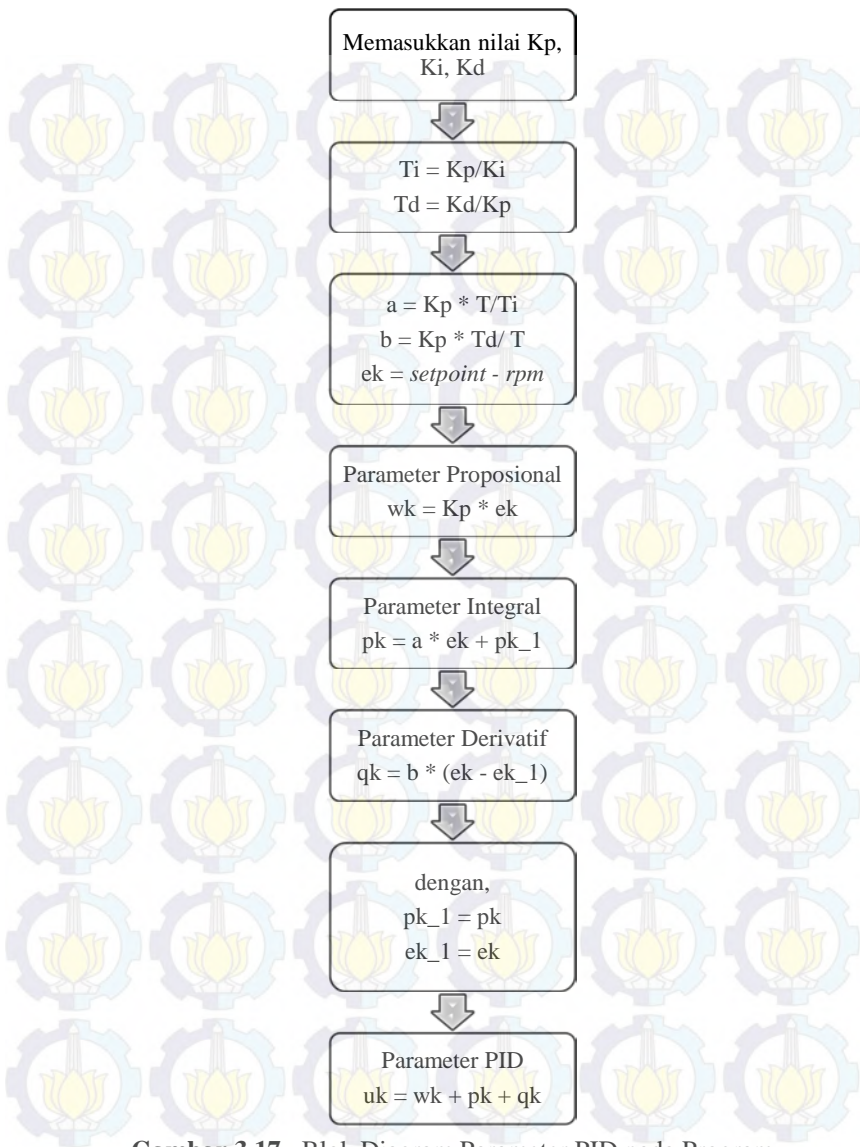
Komunikasi serial dibutuhkan untuk melakukan interaksi antara Arduino UNO dengan komputer (dalam hal ini LabVIEW). Program komunikasi terdiri dari pengaturan *baudrate* atau kecepatan transmisi data. Pengaturan *baudrate* dilakukan dengan *Serial.begin()* yang berada pada bagian *void setup()*. Beberapa pilihan kecepatan komunikasi data yang dapat digunakan pada *board* Arduino UNO adalah 300, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600 atau 115200. Berikut Gambar 3.16 berupa program yang dipergunakan untuk melakukan komunikasi serial :

```
sketch_may11a $  
void setup() {  
  Serial.begin(9600);  
}  
  
void loop()  
{  
  if (Serial.available())  
  {  
    //program  
  }  
}
```

Gambar 3.16 Program Dasar Komunikasi Serial

### 3.3.1.3. Program PID

Sistem kontrol PID terdiri dari tiga buah cara pengaturan yaitu kontrol P (*Proportional*), I (*Integral*) dan D (*Derivative*). Dalam implementasinya masing-masing cara dapat bekerja sendiri maupun gabungan. Dalam perancangan sistem kontrol PID yang perlu dilakukan adalah mengatur parameter *setpoint*, Kp, Ki dan Kd. Pembuatan program PID berdasarkan Persamaan 2.17 hingga Persamaan 2.22 mengenai persamaan PID. Berikut Gambar 3.17 yaitu blok diagram parameter PID pada program.



**Gambar 3.17** Blok Diagram Parameter PID pada Program



Berikut ini Gambar 3.18 yaitu program yang dibuat untuk melakukan pengontrolan PID:

```
read_labview_gagal
//Mengembalikan nilai asli
sp = sp_1 * 255 + sp_2;
Kp = Kp1 / 10;
Ki = Ki1 / 10;
Kd = Kd1 / 10;

Ti = Kp / Ki;
Td = Kd / Kp;

a = Kp * T / Ti;
b = Kp * Td / T;
ek = sp - rpm;

//proportional
wk = Kp * ek;
//integral
pk = a * ek + pk_1;
//derivatif
qk = b * (ek - ek_1);

//update variable
pk_1 = pk;
ek_1 = ek;

//output
uk = wk + pk + qk;
```

**Gambar 3.18** Program PID

#### **3.3.1.4. Program Konversi Desimal Menjadi Biner**

Arduino UNO memiliki kemampuan untuk menghasilkan keluaran dalam bentuk PWM atau digital dan menerima masukan dalam bentuk analog maupun digital. Arduino UNO (1) dan Arduino UNO (2) yang digunakan dalam Tugas Akhir ini saling mengirim dan menerima data. Arduino UNO (1) mengirim data dalam bentuk digital biner 8 bit dan Arduino UNO (2) menerima data dalam bentuk sinyal analog. Keluaran PID Arduino UNO (1) yang masih dalam bentuk desimal perlu dikonversi menjadi data digital biner 8 bit. Data biner tersebut akan masuk ke rangkaian DAC 8bit untuk diubah menjadi sinyal analog sehingga Arduino UNO (2) dapat menerima sinyal tersebut melalui pin ADC (A3). Berikut Gambar 3.19 berupa program konversi desimal menjadi biner 8 bit:

```

read_labview_gagal
//mengeluarkan output biner 8 bit
out = map(int(uk), 0, 2500, 0, 255);
for (int i = 0; i < 8; i++)
{
    if (out % 2 == 1)
    {
        digitalWrite(ledpins[i], HIGH);
        out = (out - 1) / 2;
    }
    else
    {
        digitalWrite(ledpins[i], LOW);
        out = out / 2;
    }
}

```

**Gambar 3.19** Program Konversi Desimal Menjadi Biner

### 3.3.1.5. Program Pengendali *Driver TRIAC*

Arduino UNO (2) digunakan untuk mengendalikan *driver TRIAC* menggunakan metode *switching*. Arduino UNO (2) menerima sinyal *interrupt* dan sinyal analog. Sinyal *interrupt* digunakan untuk mengetahui kapan program akan dimulai dan sinyal analog digunakan untuk mengetahui lama *delay switching*. Berikut Gambar 3.20 berupa program yang digunakan untuk mengendalikan *driver TRIAC* :

```

uji_coba_driver
void setup() {
    pinMode(7, OUTPUT);
    digitalWrite(7, LOW);
    attachInterrupt(0, zcd, RISING);
}

void zcd(){
    int tunda=map(analogRead(A3),0,1023,9600,300);
    delayMicroseconds(tunda);
    digitalWrite(7, HIGH);
    delayMicroseconds(50);
    digitalWrite(7, LOW);
}

void loop() {
    // put your main code here, to run repeatedly:
}

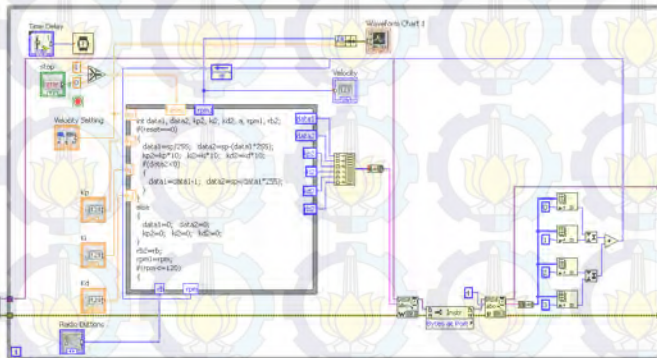
```

**Gambar 3.20** Program Pengendali *Driver TRIAC*

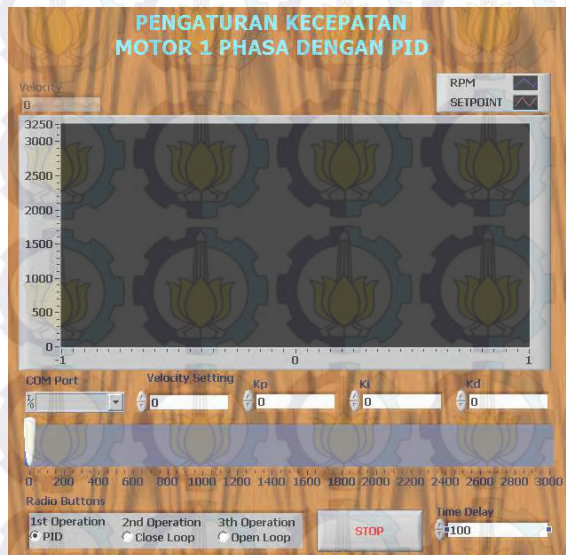
### 3.3.2. Pemrograman LabVIEW

Pemrograman LabVIEW merupakan pemrograman grafis dimana dalam Tugas Akhir ini digunakan untuk mengirim data diantaranya : Kp, Ki, Kd, dan *setpoint*. Selain itu juga dipergunakan untuk menampilkan data kecepatan motor yang terbaca oleh sensor dan

telah diproses menjadi suatu nilai rpm (rotasi per menit) oleh Arduino UNO. Berikut Gambar 3.21 berupa tampilan blok diagram dan Gambar 3.22 berupa *front panel* pemrograman LabVIEW :



**Gambar 3.21** Blok Diagram Pemrograman LabVIEW



**Gambar 3.22** Front Panel Pemrograman LabVIEW





## BAB IV

### PENGUJIAN DAN ANALISA ALAT

Pada bab ini akan dibahas mengenai pengujian dan analisa alat dari modul pengaturan motor induksi satu fasa yang telah dibuat. Pengujian alat ini ditujukan untuk memastikan agar peralatan yang dibuat dapat berfungsi dengan baik.

Pengujian ini meliputi pengujian *output* DAC 8 bit, pengujian *zero crossing detector*, pengujian *driver* TRIAC, pengujian sensor *rotary encoder* dan pengujian program PID. Setelah melakukan beberapa pengujian alat, data yang diperoleh akan dianalisa untuk mengetahui proses kerja dari seluruh sistem alat yang dibuat.

#### 4.1 Pengujian Power Supply

Pengujian *power supply* bertujuan untuk mengetahui tegangan terukur telah sesuai dengan tegangan *output power supply*. Untuk mendapatkan data tegangan terukur menggunakan tanpa beban dan berbeban pada Tabel 4.1 sebagai berikut

Tabel 4.1 Hasil Pengujian *Power Supply*

Output Power Supply (VDC)	Tegangan Terukur		Penggunaan
	Tanpa Beban (VDC)	Berbeban (VDC)	
5	4,98	4,97	ZCD, <i>Driver</i> TRIAC
8	7,98	7,98	Arduino UNO
12	11,98	11,98	DAC 8bit
-12	-12,02	-12,02	DAC 8bit

Tabel 4.1 menunjukkan bahwa hasil antara tegangan *output power supply* dengan tegangan terukur memiliki drop tegangan  $\pm 0,02$  Volt.

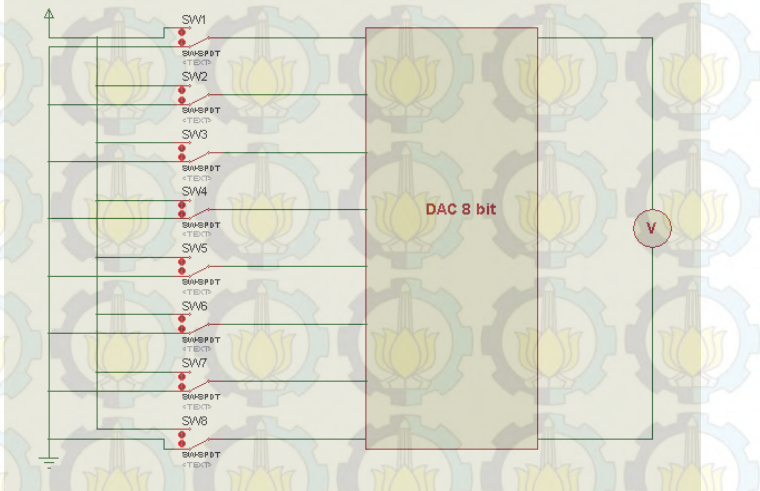
#### 4.2 Pengujian DAC 8 bit

Pengujian DAC 8 bit dilakukan untuk memperoleh data yang valid antara *input* dan *output*. Pengujian DAC 8 bit dengan mengatur  $V_{ref}$  sebesar 5 volt kemudian memberikan *input* digital dalam bentuk biner dari 0 hingga 255 menggunakan saklar SPDT seperti pada Gambar 4.1. Keluaran DAC 8 bit diukur menggunakan AVO meter SANWA CD800a. Berikut Tabel 4.2 adalah hasil pengujian DAC 8 bit sebagai berikut:

Tabel 4.2 Hasil Pengujian DAC 8 bit

Input		Output		Error (Pengukuran-Perhitungan)
Biner	Desimal	Pengukuran	Perhitungan	
0000 0000	0	0,004 V	0 V	0,004 V
0000 0001	1	0,024 V	0,019 V	0,005 V
0000 0011	3	0,062 V	0,058 V	0,004 V
0000 0111	7	0,141 V	0,136 V	0,005 V
0000 1111	15	0,297 V	0,292 V	0,005 V
0001 1111	31	0,608 V	0,605 V	0,003 V
0011 1111	63	1,233 V	1,23 V	0,003 V
0111 1111	127	2,485 V	2,48 V	0,005 V
1111 1111	255	4,98 V	4,98 V	0 V

Tabel 4.2 menunjukkan bahwa pengujian DAC 8 bit terdapat *error*  $\pm 0,005$  V pada setiap perubahan 1 bit.

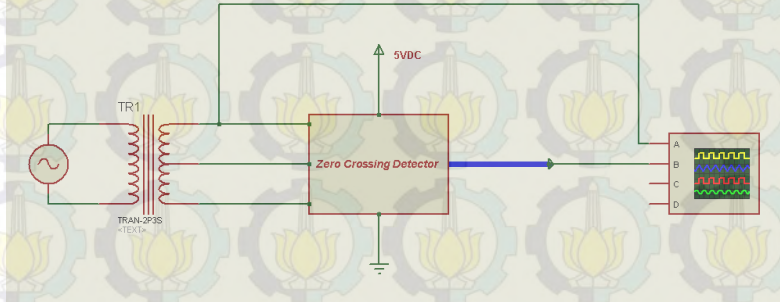


Gambar 4.1 Wiring Pengujian DAC 8 bit

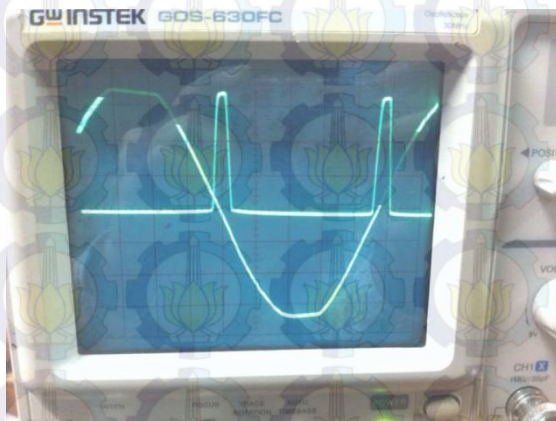
### 4.3 Pengujian Zero Crossing Detector

Pengujian *zero crossing detector* dimaksudkan untuk mengetahui apakah rangkaian *zero crossing detector* tersebut menghasilkan sinyal ketika tegangan jala-jala PLN berada pada persilangan nol seperti pada Gambar 4.3. Pengujian seperti pada Gambar 4.2 dilakukan dengan membandingkan sinyal AC tegangan jala-jala PLN yang telah

diturunkan menggunakan *transformator step down* terhadap sinyal keluaran *zero crossing detector* menggunakan osiloskop laboratorium Elektronika Terapan D3 Teknik Elektro. Berikut ini *wiring* pengujian dan hasil pengujian rangkaian *zero crossing detector* :



**Gambar 4.2** Wiring Pengujian Zero Crossing Detector



**Gambar 4.3** Hasil Pengujian Zero Crossing Detector

#### 4.4 Pengujian Driver TRIAC

Tujuan *driver TRIAC* adalah untuk menguji *driver* dapat bekerja dengan baik atau tidak ketika digunakan. Pengujian *driver* dilakukan dengan cara mengukur nilai tegangan pada beban ketika dilakukan pemotongan sinyal pada sudut-sudut tertentu. Alat ukur yang digunakan



dalam pengukuran ini adalah AVO meter SANWA CD800a. Berikut Tabel 4.3 hasil pengujian *driver* TRIAC sebagai berikut :

Tabel 4.3 Hasil Pengujian *Driver* TRIAC

Sudut Pemotongan	<i>Output</i>
0%	229 V
25%	180 V
50%	116 V
75%	60 V
±95%	6,6 V

\**Input* = 229 Volt

Tabel 4.3 menunjukkan hasil pemotongan sinyal *driver* TRIAC dengan input 229 volt pada sudut pemotongan ±95% menghasilkan keluaran 6,6 Volt.

#### 4.5 Pengujian *Rotary Encoder* dan Program Pembaca

Tujuan pengujian *rotary encoder* adalah untuk mengetahui keluaran dari *rotary encoder*. Pengujian keluaran *rotary encoder* dilakukan dengan cara memasang *rotary encoder* ke bagian kopel antara sensor dengan motor sehingga *rotary encoder* akan terhubung dengan rotor motor AC. Pengujian sinyal keluaran *rotary encoder* dilakukan dengan cara mengamati osiloskop. Osiloskop yang digunakan adalah osiloskop Laboratorium Elektronika Terapan D3 Teknik Elektro. Kemudian dilakukan pengujian program pembacaan *rotary encoder*. Tujuan dari pengujian program pembacaan *rotary encoder* ini adalah untuk mengetahui apakah program yang dibuat sudah dapat diaplikasikan atau belum sehingga nantinya hasil *rotary encoder* dari program yang masuk akan sesuai dengan data *rotary encoder* pada motor yang sebenarnya. Pengujian program dilakukan dengan cara membandingkan antara pembacaan sensor *rotary encoder* dengan tachometer merk Dekko Laboratorium Elektronika Terapan D3 Teknik Elektro. Dari data pengujian dan pengukuran didapatkan nilai *error* sebagai berikut :

$$Error = \text{Pengukuran} - \text{Pengujian} \quad (4.1)$$

Keterangan :

Pengujian = Nilai dari *rotary encoder* yang telah dikalibrasi



Pengukuran = Nilai yang terukur menggunakan tachometer

Pada Gambar 4.4 dapat dicari nilai kecepatan menggunakan Persamaan 3.1 untuk mengubah jumlah pulsa menjadi nilai kecepatan seperti dibawah ini :

$$f = 1,2593 \text{ KHz} = 1259$$

$$T = 1 \text{ detik}$$

$$ppr = 25$$

$$n = \frac{60 \cdot f}{T \cdot ppr}$$

$$n = \frac{60 \cdot 1259}{1.25}$$

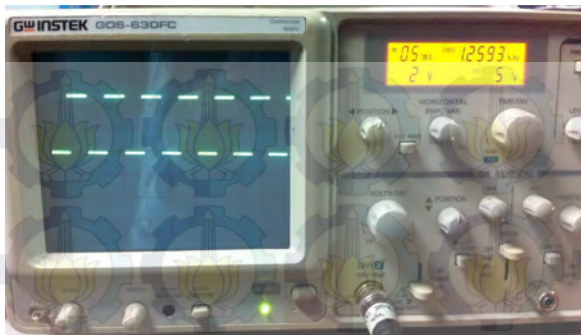
$$n = \frac{75540}{25}$$

$$n = 3024$$

dalam pengujian ini mikrokontroler digunakan untuk mengubah jumlah pulsa menjadi nilai kecepatan menggunakan persamaan 3.1 seperti contoh diatas. Pada Gambar 4.5 diperlihatkan hasil pembacaan kecepatan oleh *rotary encoder*. Pada Tabel 4.4 diperlihatkan hasil pengujian program pembacaan *rotary encoder* sebagai berikut :

Tabel 4.4 Pengujian Program Pembacaan *Rotary Encoder*

Pengukuran (rpm)	Pengujian (rpm)	Error (pengukuran - pengujian)
1008	1024	-16
1274	1249	25
1531	1572	-41
1754	1798	-44
1999	2000	-1
2256	2226	30
2507	2548	-41
2754	2774	-20
2998	3024	-26



**Gambar 4.4** Pengujian Keluaran Rotary Encoder



**Gambar 4.5** Pengujian Program Pembacaan Rotary Encoder

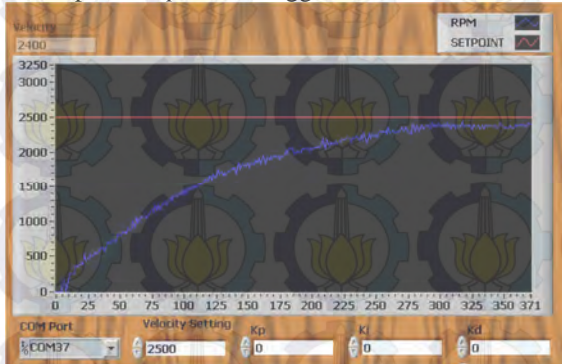
#### 4.6 Pengujian Kontrol PID

Pengujian motor menggunakan metode kontrol PID. Tahap pengujian respon motor ini dilakukan tiga pengujian yaitu pengujian nilai  $K_p$ ,  $K_i$ ,  $K_d$

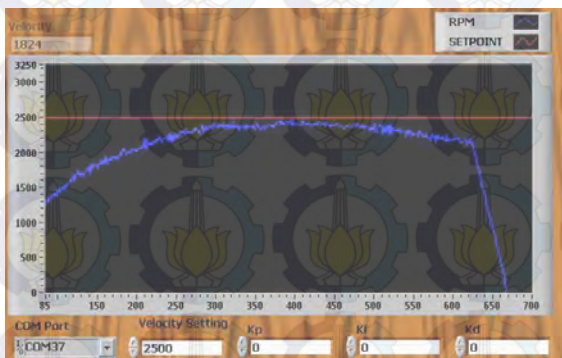
##### 4.5.1 Pengujian Respon Motor Tanpa Beban dan Berbeban pada Sistem *Open Loop*

Pengujian ini bertujuan melihat respon motor menggunakan sistem *open-loop* pada saat *setpoint* bernilai 2500 rpm. Dimana, metode ini memiliki 2 parameter yaitu  $T_D$  (waktu mati) dan  $T_1$  (waktu tunda). Maka dari Gambar 4.6 menunjukkan grafik motor tanpa beban yang

stabil saat kecepatan 2400 rpm. Sehingga didapatkan nilai  $T_D$  sebesar 0 detik,  $T_1$  sebesar 18,5 detik dan *rise time* 30 detik. Sedangkan pada Gambar 4.7 menunjukkan grafik motor berbeban pada kecepatan 2400 rpm saat detik ke 66 motor diberi beban dan sistem tidak dapat mengembalikan pada *set point* sehingga motor berhenti.



**Gambar 4.6** Grafik Respon Motor Sistem *Open Loop* Tanpa Beban



**Gambar 4.7** Grafik Respon Motor Sistem *Open Loop* Berbeban

#### 4.5.2 Pengujian Respon Motor Terhadap Sistem PID

Pengujian ini dilakukan untuk melihat respon motor dalam mencapai nilai kecepatan referensi atau *setpoint*. Respon motor dalam mencapai kecepatan referensi menggunakan sistem PID dengan pengujian berulang-ulang atau *trial & error* untuk mendapatkan nilai  $K_p$ ,  $K_i$ ,  $K_d$  pada kecepatan referensi 2500 rpm seperti yang terdata pada Tabel 4.5 hasil pengujian nilai  $K_p$  sebagai berikut :



Tabel 4.5 Pengujian Nilai Kp

Kp	Ki	Kd	Steady State	Rise Time(s)	Settling Time(s)
1,7	0	0	1152	0,5	1,5
2	0	0	1344	0,9	1,7
2,3	0	0	1488	1,1	2

Tabel 4.5 menunjukkan bahwa semakin tinggi nilai Kp sehingga semakin cepat *steady state* yang dapat dicapai motor. Nilai parameter Kp menggunakan Kp 2,3 didapatkan dari metode *trial and error*. Sedangkan nilai Kp 1,7 dan Kp 2 sebagai perbandingan respon dari Kp 2,3. Tabel 4.5 dan Gambar 4.7 hingga Gambar 4.9 menunjukkan grafik saat Kp bernilai 2,3 motor memiliki *steady state* yang lebih tinggi dibandingkan dengan Kp 1,7 dan Kp 2. Maka nilai Kp 2,3 digunakan sebagai kontrol proposional dalam sistem PID. Hal ini dikarenakan kontrol proposional berfungsi untuk memperkuat sinyal sehingga mempercepat respon mencapai *set point*.

Setelah mendapatkan nilai Kp yang stabil untuk kecepatan 2500rpm maka dilanjutkan mencari nilai Ki saat nilai Kp 2,3. Berikut Tabel 4.6 hasil pengujian nilai Kp dan Ki sebagai berikut :

Tabel 4.6 Pengujian Nilai Kp dan Ki

Kp	Ki	Kd	Steady State	Rise Time(s)	Settling Time(s)
2,3	1	0	2500	7,6	8,9
2,3	1,7	0	2500	4,2	6,3
2,3	2,2	0	2500	4,5	6,5

Pada Tabel 4.6 nilai parameter Ki menggunakan Ki 1,7 didapatkan dari metode *trial and error*. Sedangkan nilai Ki 1 dan Ki 2,2 sebagai perbandingan respon dari Ki 1,7. Tabel 4.6 dan Gambar 4.10 hingga Gambar 4.12 menunjukkan bahwa saat Ki semakin besar bukan berarti respon motor semakin cepat maka nilai Ki 1,7 digunakan sebagai kontrol integral dalam sistem PID dengan nilai *rise time* dan *settling time* Ki 1,7 lebih kecil daripada Ki 1 dan Ki 2,2.

Setelah mendapatkan nilai Ki yang stabil untuk kecepatan 2500rpm maka dilanjutkan mencari nilai Kd saat nilai Kp 2,3 dan Ki

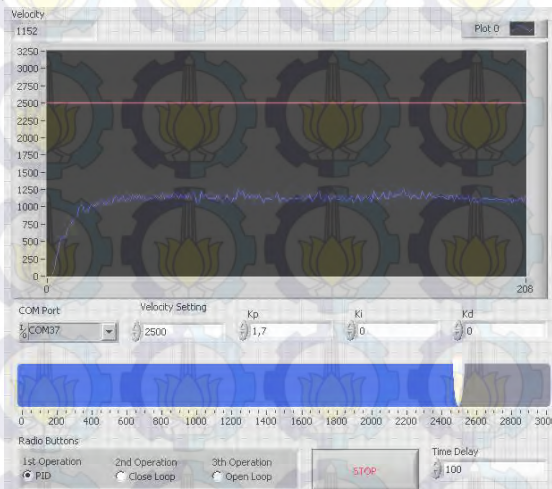


1,7. Berikut Tabel 4.7 hasil pengujian nilai Kp, Ki, dan Kd sebagai berikut :

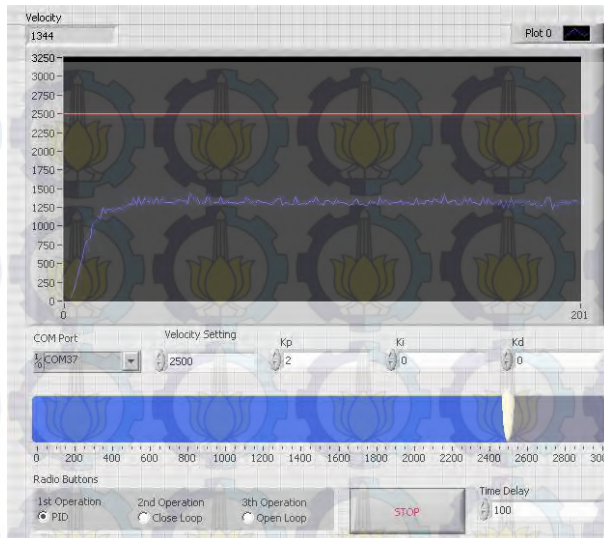
Tabel 4.7 Pengujian Nilai Kp, Ki, dan Kd

Kp	Ki	Kd	Steady State	Rise Time(s)	Settling Time(s)	Overshoot
2,3	1,7	0,1	2500	2,9	7	150
2,3	1,7	0,5	2500	2,9	10	250
2,3	1,7	2,2	2500	3	12,8	400

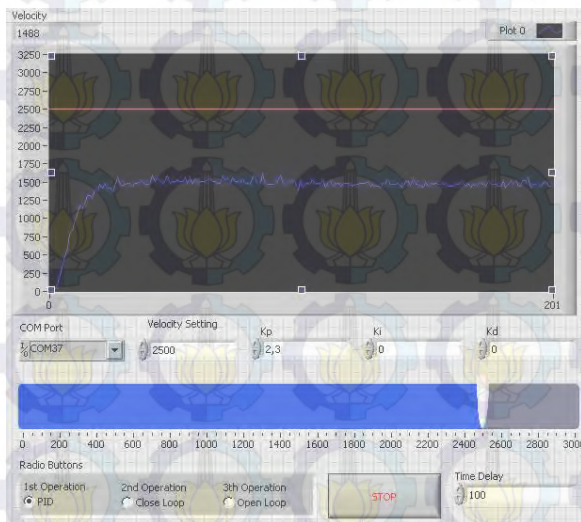
Pada Tabel 4.7 nilai parameter Kd menggunakan Kd 0,1 didapatkan dari metode *trial and error*. Sedangkan nilai Kd 0,5 dan Kd 2,2 sebagai perbandingan respon dari Kd 0,1. Tabel 4.7 dan Gambar 4.13 hingga Gambar 4.15 menunjukkan bahwa perubahan nilai Kd memiliki nilai *rise time* yang hampir sama. Maka nilai Kd 0,1 digunakan sebagai kontrol derivatif dalam sistem PID dikarenakan jika semakin besar Kd maka semakin besar nilai *settling time* dan *overshoot* yang dihasilkan.



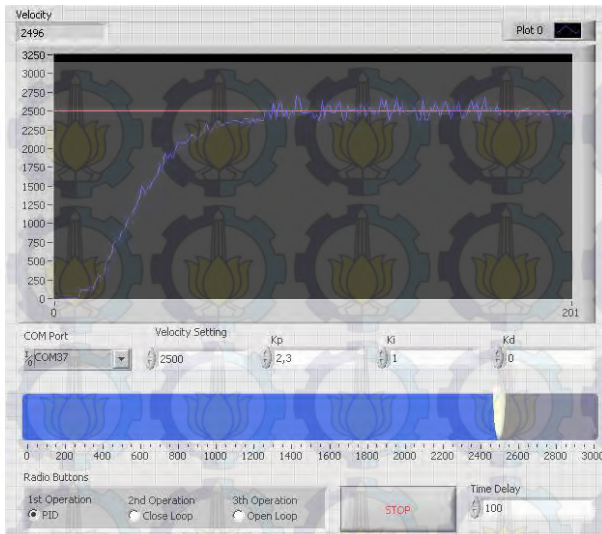
Gambar 4.8 Grafik Respon Motor pada Nilai Kp 1,7



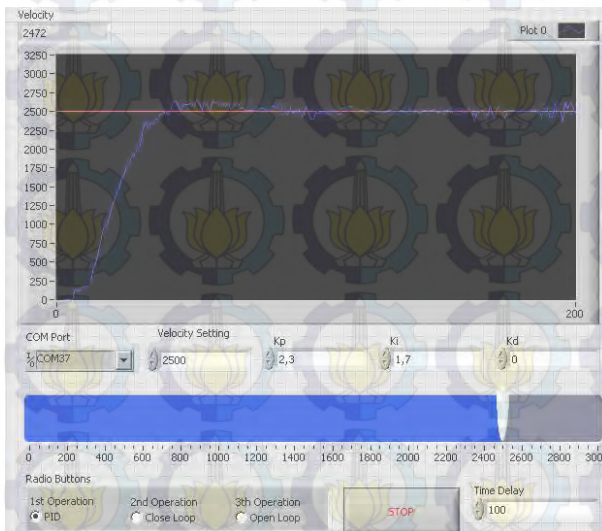
**Gambar 4.9** Grafik Respon Motor pada Nilai Kp 2



**Gambar 4.10** Grafik Respon Motor pada Nilai Kp 2,3

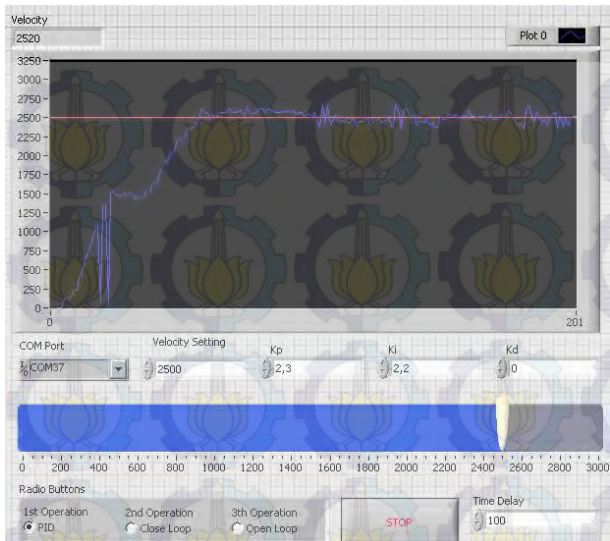


**Gambar 4.11** Grafik Respon Motor pada Nilai Ki 1

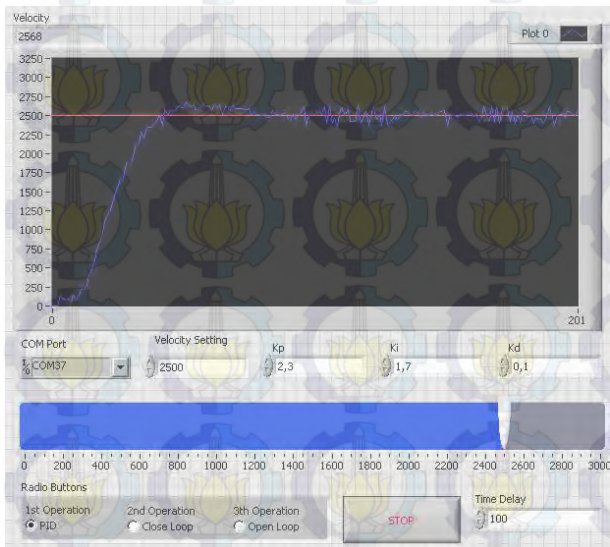


**Gambar 4.12** Grafik Respon Motor pada Nilai Ki 1,7



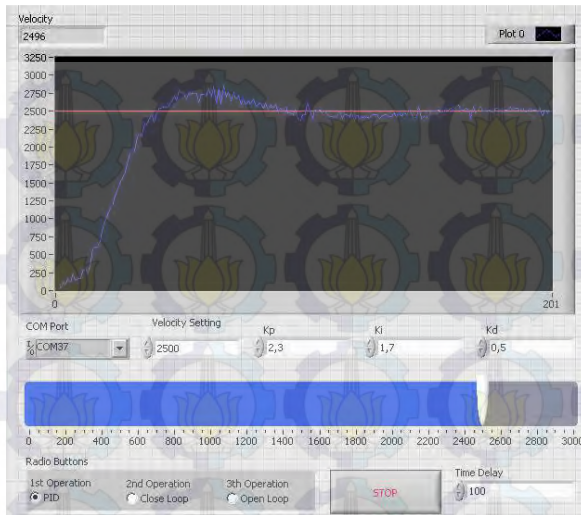


**Gambar 4.13** Grafik Respon Motor pada Nilai Ki 2,2

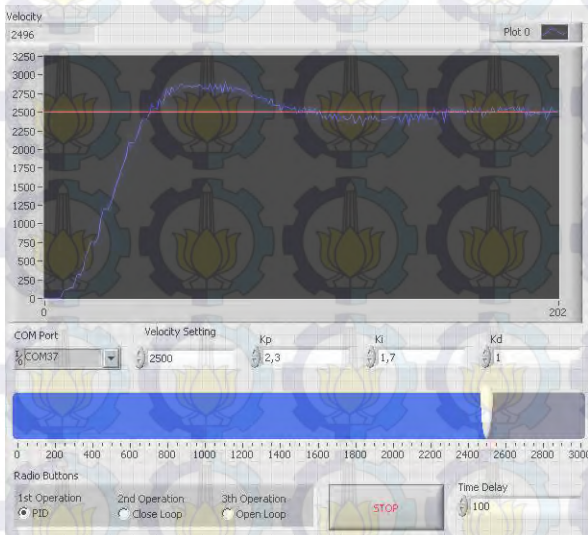


**Gambar 4.14** Grafik Respon Motor pada Nilai Kd 0,1





**Gambar 4.15** Grafik Respon Motor pada Nilai Kd 0,5



**Gambar 4.16** Grafik Respon Motor pada Nilai Kd 1

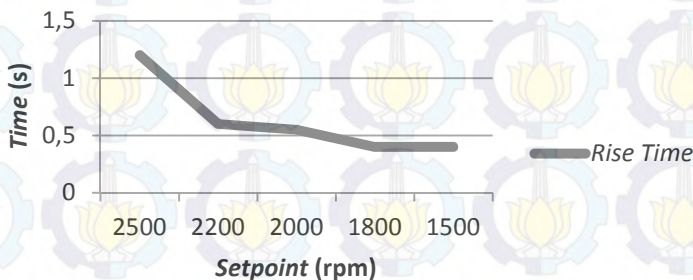
### 4.5.3 Pengujian Respon Motor terhadap *Rise Time*

Pengujian ini dilakukan untuk melihat respon motor dalam keadaan tanpa beban mencapai nilai kecepatan referensi atau *setpoint*. Berikut Tabel 4.8 hasil pengujian nilai *rise time* respon motor tanpa beban sebagai berikut :

Tabel 4.8 Pengujian *Rise Time* Respon Motor Tanpa Beban

Kecepatan (rpm)	<i>Rise Time</i> (s)	<i>Settling Time</i> (s)
2500	1,2	4
2200	0,6	9
2000	0,55	9
1800	0,4	12
1500	0,4	15

Tabel 4.8 menunjukkan bahwa semakin kecil nilai *setpoint* kecepatan motor maka semakin cepat respon motor mencapai nilai *rise time*. Karena, sistem pengendali lebih cepat mengatur kecepatan pada kecepatan referensi yang lebih rendah daripada kecepatan referensi yang lebih tinggi. Gambar 4.16 menunjukkan perbandingan antara kecepatan referensi motor dengan *rise time*. Semakin besar nilai kecepatan referensi maka semakin lama waktu *rise time* motor. Hal ini dikarenakan, pada sistem kendali memerlukan waktu untuk mengatur motor pada kecepatan referensinya.



Gambar 4.17 Grafik Perbandingan *Rise Time* Terhadap *Setpoint*

### 4.5.4 Pengujian Respon Motor Berbeban

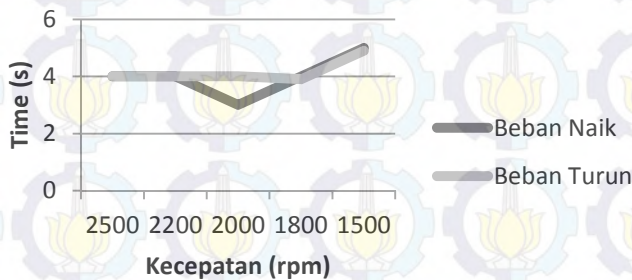
Pengujian respon motor berbeban ini dilakukan dengan memberi beban mekanik pada motor sehingga dapat diketahui respon motor untuk

mencapai kembali kecepatan referensi atau *setpoint*. Dimana beban naik ialah hambatan mekanik yang diberikan saat motor berputar, sedangkan beban turun ialah saat hambatan mekanik dilepas. Berikut Tabel 4.9 hasil pengujian *steady state* motor berbeban sebagai berikut:

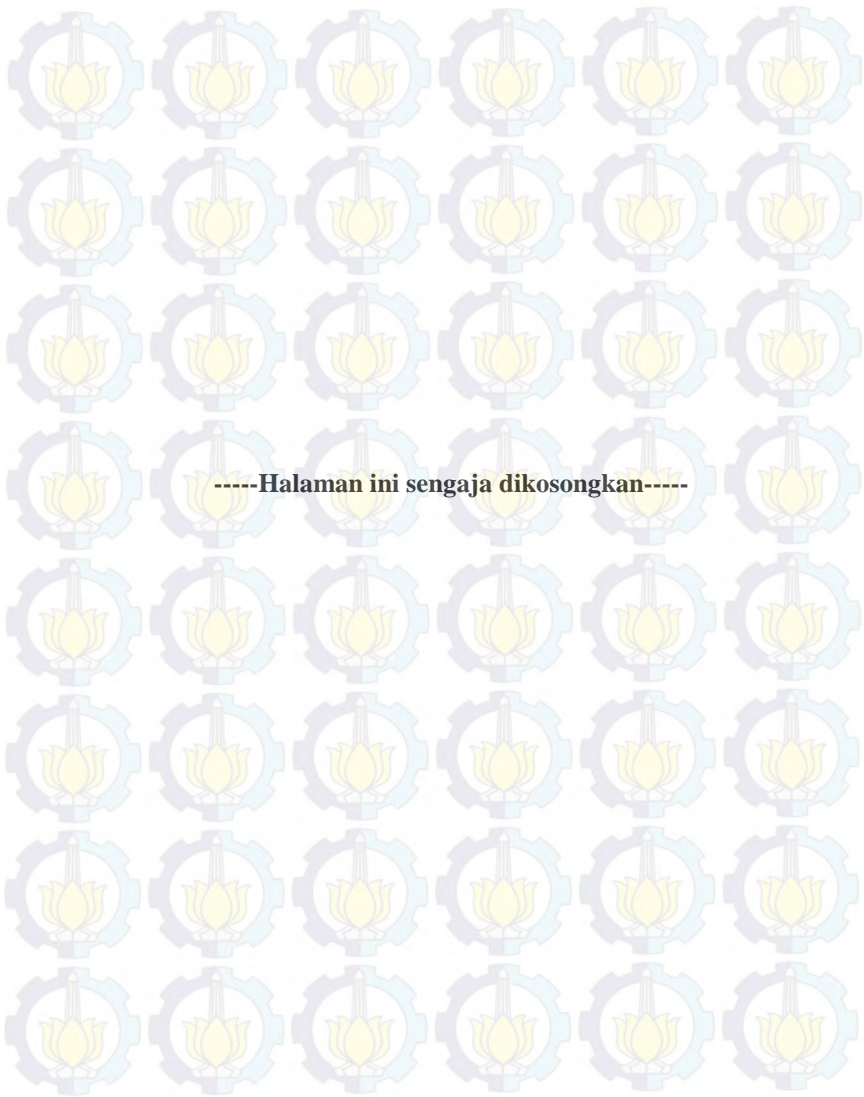
Tabel 4.9 Pengujian *Steady State* Berbeban

Kecepatan (rpm)	Settling Time (s)	
	Beban naik	Beban turun
2500	4	4
2200	4	4
2000	3	4
1800	4	3,9
1500	5	4,9

Tabel 4.9 menunjukkan bahwa pada nilai *steady state* akan berbeda pada setiap perubahan kecepatan. Hal ini, dikarenakan alat pengendali membutuhkan waktu untuk menghitung dan mengembalikan putaran pada kecepatan referensi saat adanya hambatan atau beban. Gambar 4.17 menunjukkan grafik perbandingan *steady state* pada saat beban naik dengan beban turun disetiap perubahan kecepatannya. Terjadi perbedaan waktu  $\pm 0,1$  detik saat beban naik maupun beban turun untuk mencapai nilai *steady state*.



**Gambar 4.18** Grafik Perbandingan *Settling Time* Saat Beban Naik dan Turun





## **BAB V**

### **PENUTUP**

Bab penutup ini berisi tentang kesimpulan dan saran yang diperoleh selama proses pembuatan modul pengaturan kecepatan motor induksi satu phasa

#### **5.1 Kesimpulan**

1. Motor dengan sistem *open loop* menggunakan *setpoint* 2500 rpm tanpa beban didapatkan nilai *rise time* 30 detik. Sedangkan motor pada sistem *close loop* dengan PID menggunakan *setpoint* yang sama didapatkan nilai *rise time* 1,2 detik.
2. Ketika motor dengan sistem *open loop* menggunakan *setpoint* 2500 rpm diberikan beban, kecepatan motor turun hingga motor berhenti. Sedangkan motor pada sistem *close loop* dengan PID menggunakan *setpoint* yang sama, kecepatan motor juga turun akan tetapi dapat kembali ke nilai *setpoint*-nya dengan nilai *rise time* 4 detik.
3. Perbedaan antara sistem *open loop* dengan sistem *close loop* PID adalah pada sistem *open loop* tidak terdapat umpan balik sedangkan pada sistem *close loop* dengan PID terdapat umpan balik berupa nilai kecepatan motor.

#### **5.2 Saran**

1. Untuk mendapatkan sistem pengaturan kecepatan yang lebih stabil dan mudah diperlukan perbandingan dengan metode yang berbeda.
2. Untuk melihat sistem pengaturan kecepatan dapat bekerja akurat dibutuhkan motor dengan daya yang lebih besar.



## DAFTAR PUSTAKA

- [1] Sinaga, Ridwan. “*Analisis Karakteristik Berbeban Motor Induksi Satu Fasa Kapasitor Start*”, Tugas Akhir. Universitas Sumatera Utara , 2011
- [2] Wirawan, Giri Nata. (2015). “*Sistem Instrumentasi Elektronika.*” <http://dokumen.tips/documents/zero-crossing.html>[10 Mei 2016]
- [3] Morris, Alan S., “*Measurement and Instrumentation Principles 3rd Edition*”, Oxford: Butterworth-Heinemann, 2001. hal. 392-394
- [4] Satria, Indera Surya. “*Monitoring dan Pendeteksian Lokasi Kebocoran Pipa Saluran Air Bersih Menggunakan Flow Liquid Meter Sensor pada Sensor Network*”, Tugas Akhir. Universitas Sumatera Utara , 2016
- [5] Sigiro, Stevani Agnesia. (2015). “*Instrumentasi Virtual Menggunakan Labview Dan Soundcard*”, Tugas Akhir. Universitas Sumatera Utara , 2015
- [6] Datasheet DAC0808
- [7] Wisesa, Tirta. (2014). “*Perancangan Pengaturan Kecepatan Motor Induksi Satu Fasa dengan PWM Menggunakan Pengendali PID Berbasis Arduino*”, Tugas Akhir. Universitas Bengkulu , 2014.
- [8] Ibrahim, Dogan. “*Microcontroller Based Applied Digital Control*”. England: John Wiley & Sons, Ltd, 2006.





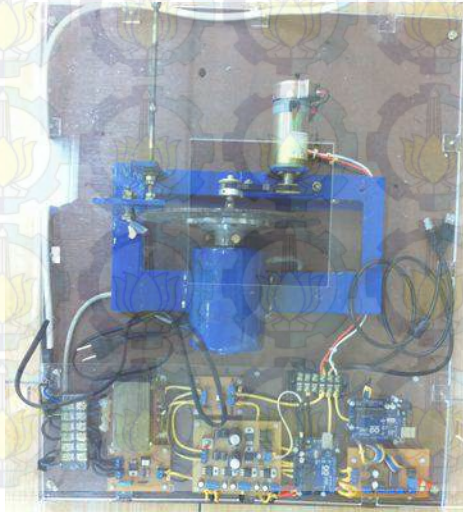
## **LAMPIRAN A**

### **FOTO ALAT**

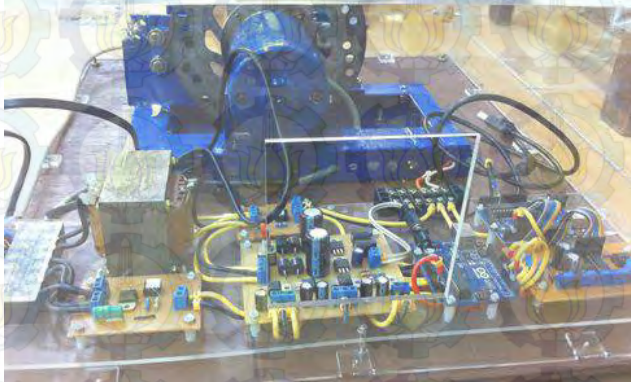
#### **1. MODUL PENGATURAN KECEPATAN MOTOR 1 PHASA MENGUNAKAN PID TAMPAK DEPAN**



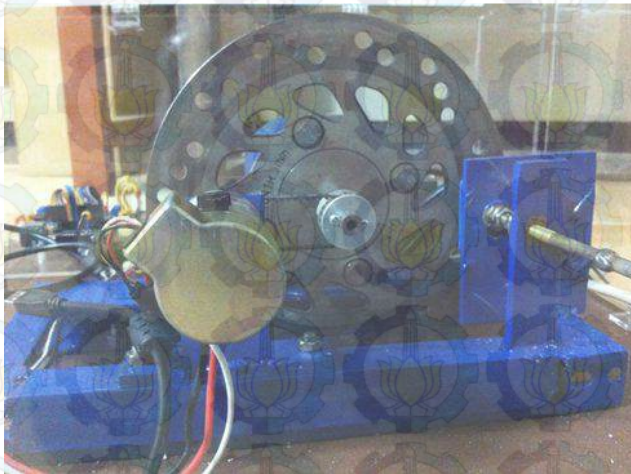
#### **2. MODUL PENGATURAN KECEPATAN MOTOR 1 PHASA MENGUNAKAN PID TAMPAK TENGAH**



**3. MODUL PENGATURAN KECEPATAN MOTOR 1 PHASA  
MENGUNAKAN PID TAMPAK BELAKANG**



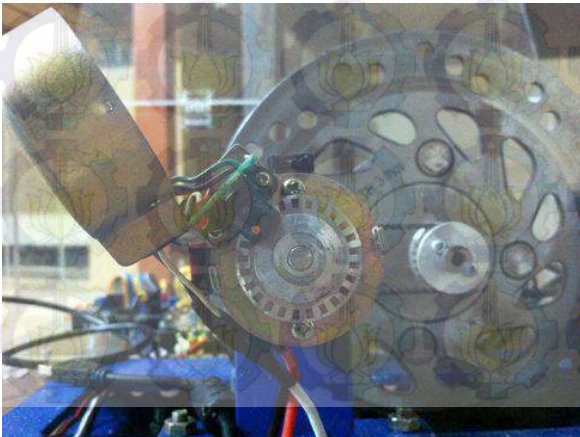
**4. DESAIN *ROTARY ENCODER* TAMPAK DEPAN**



**5. DESAIN *ROTARY ENCODER* TAMPAK ATAS**



**6. DESAIN *ROTARY ENCODER* BAGIAN DALAM TAMPAK DEPAN**





## 7. DESAIN *ROTARY ENCODER* BAGIAN DALAM TAMPAK SAMPING





## LAMPIRAN B PROGRAM

### 1. Program Arduino 1

```
#include <TimerOne.h>

int ledpins[ ] = {3, 4, 5, 6, 7, 8, 9, 10};
unsigned int counter = 0;

int out;
int rb;

float rpm, rpm1, rpm2, sp, sp_1, sp_2, uk, Kp, Ki, Kd, Kp1, Ki1, Kd1,
ek, Ti, Td, a, b, wk, pk, qk;

float pk_1 = 0;
float ek_1 = 0;

const float T = 0.1;

void setup()
{
  Serial.begin(9600);
  for (int i = 0; i < 8; i++)
  {
    pinMode(ledpins[i], OUTPUT);
  }

  Timer1.initialize(100000);
  attachInterrupt(0, docount, RISING);
  Timer1.attachInterrupt( timerIsr );
}

void docount()
{
  counter++;
}
```

```

void loop()
{
}

void timerIsr()
{
  Timer1.detachInterrupt();

  rpm = ((counter * 60) / (25 * 0.1));

  if (Serial.available() > 0)
  {
    delay(10);
    sp_1 = Serial.read();
    sp_2 = Serial.read();
    Kp1 = Serial.read();
    Ki1 = Serial.read();
    Kd1 = Serial.read();
    rb = Serial.read();

    //Mengembalikan nilai asli
    sp = sp_1 * 255 + sp_2;
    Kp = Kp1 / 10;
    Ki = Ki1 / 10;
    Kd = Kd1 / 10;

    switch (rb)
    {
      case 0:
        if (Ki == 0)
        {
          Ti = 0;
        }
        else
        {
          Ti = Kp / Ki;
        }
        if (Kp == 0)

```

```

{
    Td = 0;
}
else
{
    Td = Kd / Kp;
}

if (Ti == 0)
{
    a = 0;
}
else
{
    a = Kp * T / Ti;
}
b = Kp * Td / T;
ek = sp - rpm;

//proportional
wk = Kp * ek;
//integral
pk = a * ek + pk_1;
//derivatif
qk = b * (ek - ek_1);

//update variable
pk_1 = pk;
ek_1 = ek;

//output
uk = wk + pk + qk;
break;

case 1:
    uk = sp - ek;
    break;

```

case 2:

```

    uk = sp;
    break;
}

if(sp=0)
{
    uk=0;
    wk=0;
    pk=0;
    qk=0;
    pk_1=0;
    ek_1=0;
    uk=0;
}

if (uk <= 0)
{
    uk = 0;
}

//mengeluarkan output biner 8 bit
out = map(int(uk), 0, 5900, 0, 255);
for (int i = 0; i < 8; i++)
{
    if (out % 2 == 1)
    {
        digitalWrite(ledpins[i], HIGH);
        out = ((out - 1) / 2);
    }
    else
    {
        digitalWrite(ledpins[i], LOW);
        out = out / 2;
    }
}

if (rpm >= 2500)
{
    rpm1 = 2500;

```



```

    rpm2 = rpm - 2500;
}
else
{
    rpm1 = rpm;
    rpm2 = 0;
}
delay(10);
Serial.write(highByte(int(rpm1)));
Serial.write(lowByte(int(rpm1)));
delay(10);
Serial.write(highByte(int(rpm2)));
Serial.write(lowByte(int(rpm2)));
Serial.flush();
}
counter = 0;
Timer1.attachInterrupt( timerIsr );
}

```

## 2. Program Arduino (2)

```

void setup() {
    pinMode(7, OUTPUT);
    digitalWrite(7, LOW);
    attachInterrupt(0, zcd, RISING);
}

void zcd(){
    int tunda=map(analogRead(A3),0,1023,9600,300);
    delayMicroseconds(tunda);
    digitalWrite(7, HIGH);
    delayMicroseconds(50);
    digitalWrite(7, LOW);
}

void loop() {
    // put your main code here, to run repeatedly:
}

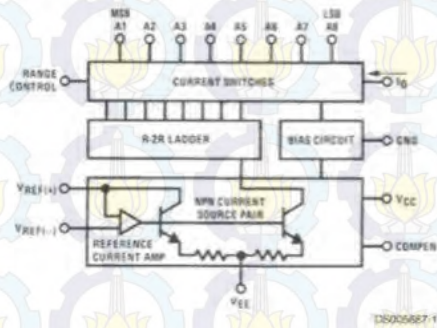
```



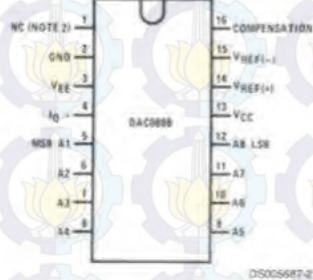
## LAMPIRAN C DATASHEET

### 1. Datasheet DAC 0808

#### Block and Connection Diagrams



#### Dual-In-Line Package



**Top View**  
Order Number DAC0808  
See NS Package M16A or N16A

# Absolute Maximum Ratings (Note 2)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Power Supply Voltage

$V_{CC}$   $+18 V_{DD}$   
 $V_{EE}$   $-18 V_{DD}$

Digital Input Voltage,  $V_{I1}$ - $V_{I12}$

$-10 V_{DD}$  to  $+18 V_{DD}$

Applied Output Voltage,  $V_{O1}$

$-11 V_{DD}$  to  $+18 V_{DD}$

Reference Current,  $I_{REF}$

5 mA

Reference Amplifier Inputs,  $V_{I4}$ ,  $V_{I5}$

$V_{CC}$ ,  $V_{EE}$

Power Dissipation (Note 4)

1000 mW

ESD Susceptibility (Note 5)

TBD

Storage Temperature Range

$-65^{\circ}\text{C}$  to  $+150^{\circ}\text{C}$

Lead Temp. (Soldering, 10 seconds)

$260^{\circ}\text{C}$

Dual-In-Line Package (Plastic)

$300^{\circ}\text{C}$

Dual-In-Line Package (Ceramic)

Surface Mount Package

Vapor Phase (60 seconds)

$215^{\circ}\text{C}$

Infrared (15 seconds)

$220^{\circ}\text{C}$

## Operating Ratings

Temperature Range

$T_{MIN} \leq T_A \leq T_{MAX}$

DACC808

$0 \leq T_A \leq +75^{\circ}\text{C}$

## Electrical Characteristics

( $V_{CC} = 5\text{V}$ ,  $V_{EE} = -15\text{V}$ ,  $V_{DD}/R14 = 2\text{mA}$ , and all digital inputs at high logic level unless otherwise noted.)

Symbol	Parameter	Conditions	Min	Typ	Max	Units
$E$	Relative Accuracy (Error Relative to Full Scale $I_{FS}$ )	(Figure 4)				%
	DACC808LG (LM1408-8)				$\pm 0.19$	%
	Setting Time to Within $\frac{1}{2}$ LSB (Includes $t_{SETT}$ )	$T_A = 25^{\circ}\text{C}$ (Note 7), (Figure 5)		150		ns
$t_{PROP}$ , $t_{SETT}$	Propagation Delay Time	$T_A = 25^{\circ}\text{C}$ , (Figure 6)		30	100	ns
$TCL$	Output Full-Scale Current Drift			$\pm 20$		ppm/ $^{\circ}\text{C}$
MSB	Digital Input Logic Levels	(Figure 8)				
$V_{IH}$	High Level, Logic "1"		2			$V_{DD}$
$V_{IL}$	Low Level, Logic "0"				0.8	$V_{DD}$
MSB	Digital Input Current	(Figure 3)				
	High Level	$V_{IH} = 5\text{V}$		0	0.040	mA
	Low Level	$V_{IL} = 0.8\text{V}$		-0.500	-0.8	mA
$I_{REF}$	Reference Input Bias Current	(Figure 3)		-1	-3	$\mu\text{A}$
	Output Current Range	(Figure 3)				
		$V_{EE} = -8\text{V}$	0	2.5	2.1	mA
		$V_{EE} = -15\text{V}$ , $T_A = 25^{\circ}\text{C}$	0	2.0	4.2	mA
$I_{OH}$	Output Current	$V_{REF} = 2.000\text{V}$ , $R14 = 1000\Omega$ , (Figure 3)	1.9	1.99	2.1	mA
	Output Current, All Bits Low	(Figure 3)		0	4	$\mu\text{A}$
	Output Voltage Compliance (Note 3)	$E \leq 0.19\%$ , $T_A = 25^{\circ}\text{C}$				
	$V_{CC} = 5\text{V}$ , $I_{OCC} = 1\text{mA}$				-0.55, +0.4	$V_{DD}$
	$V_{EE}$ Below $-10\text{V}$				-5.0, +0.4	$V_{DD}$
$SR_{REF}$	Reference Current Slew Rate	(Figure 6)	4	8		mA/ $\mu\text{s}$
	Output Current Power Supply Sensitivity	$-5\text{V} \leq V_{EE} \leq -15\text{V}$		0.05	2.7	$\mu\text{A/V}$
	Power Supply Current (All Bits Low)	(Figure 3)				
$I_{OCC}$				2.3	22	mA
$I_{REF}$				-4.3	-12	mA
$V_{CC}$	Power Supply Voltage Range	$T_A = 25^{\circ}\text{C}$ , (Figure 3)	4.5	5.0	5.5	$V_{DD}$
$V_{EE}$			-4.5	-15	-16.5	$V_{DD}$
	Power Dissipation					



# Electrical Characteristics (Continued)

( $V_{CC} = 5V$ ,  $V_{EE} = -15V$ ,  $V_{DD1}, V_{DD2}, V_{DD3}, V_{DD4} = 2mA$ , and all digital inputs at high logic level unless otherwise noted.)

Symbol	Parameter	Conditions	Min	Typ	Max	Units
	All Bits Low	$V_{CC} = 5V$ , $V_{EE} = -5V$		33	170	mW
	All Bits High	$V_{CC} = 5V$ , $V_{EE} = -15V$		106	305	mW
		$V_{CC} = 15V$ , $V_{EE} = -5V$		90		mW
		$V_{CC} = 15V$ , $V_{EE} = -15V$		160		mW

**Note 2:** Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications do not apply when operating the device beyond its specified operating conditions.

**Note 3:** Range control is not required.

**Note 4:** The maximum power dissipation must be derated at elevated temperatures and is limited by  $T_{JMAX}$ ,  $T_{JA}$ , and the ambient temperature,  $T_A$ . The maximum allowable power dissipation at any temperature is  $P_D = (T_{JMAX} - T_A)/R_{JA}$  in the number given in the Absolute Maximum Ratings, whichever is lower. For the package,  $T_{JMAX} = 125^\circ C$ , and the typical junction-to-ambient thermal resistance of the dual-in-line package when the board is mounted is  $100^\circ C/W$ . For the dual-in-line package, this number increases to  $170^\circ C/W$  and for the lead outline package this number is  $100^\circ C/W$ .

**Note 5:** Human body model,  $100pF$  capacitor through a  $1.5k\Omega$  resistor.

**Note 6:** All current switches are tested to guarantee at least 30% of rated current.

**Note 7:** All bits switched.

**Note 8:** Pin-out numbers for the OA1000A represent the dual-in-line package. The lead outline package pinout differs from the dual-in-line package.

## Typical Application

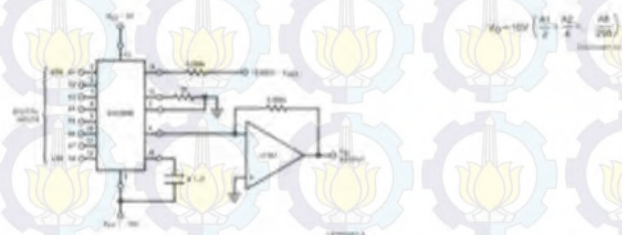
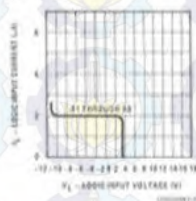


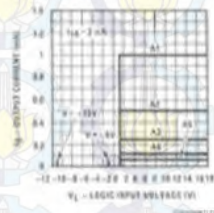
FIGURE 1. +10V Output Digital to Analog Converter (Note 8)

## Typical Performance Characteristics $V_{CC} = 5V$ , $V_{EE} = -15V$ , $T_A = 25^\circ C$ , unless otherwise noted.

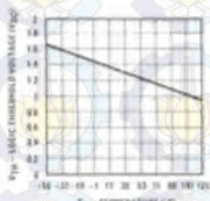
Logic Input Current vs Input Voltage



Bit Transfer Characteristics



Logic Threshold Voltage vs Temperature



## 2. Datasheet LM741



LM741

REVISED - MAY 1989 - REVISED OCTOBER 2015

### LM741 Operational Amplifier

#### 1 Features

- Overload Protection on the Input and Output
- No Latch-Up When the Common-Mode Range is Exceeded

#### 2 Applications

- Comparators
- Multivibrators
- DC Amplifiers
- Summing Amplifiers
- Integrator or Differentiators
- Active Filters

#### 3 Description

The LM741 series are general-purpose operational amplifiers which feature improved performance over industry standards like the LM709. They are direct, pin-for-pin replacements for the 709C, LM201, MC1439, and 748 in most applications.

The amplifiers offer many features which make their application nearly foolproof: overload protection on the input and output, no latch-up when the common-mode range is exceeded, as well as freedom from oscillations.

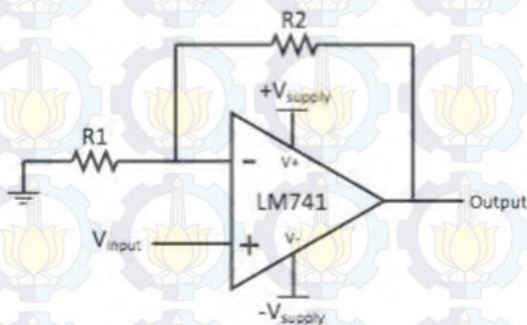
The LM741C is identical to the LM741 and LM741A except that the LM741C has their performance ensured over a 0°C to +70°C temperature range, instead of -55°C to +125°C.

#### Device Information<sup>(1)</sup>

PART NUMBER	PACKAGE	BODY SIZE (NOM)
LM741	TO-99 (B)	9.08 mm × 9.08 mm
	CDIP (B)	10.16 mm × 8.802 mm
	PDIP (B)	9.81 mm × 6.35 mm

<sup>(1)</sup> For all available packages, see the ordering addendum at the end of the data sheet.

#### Typical Application



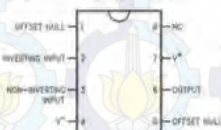
## 5 Pin Configuration and Functions

LMC Package  
8-Pin TO-99  
Top View



LM741H is available per JM3851Q/10101

NAB Package  
8-Pin CDIP or PDIP  
Top View



### Pin Functions

NAME	PIN NO.	I/O	DESCRIPTION
INVERTING INPUT	2	I	Inverting signal input
NC	8	N/A	No Connect, should be left floating
NONINVERTING INPUT	3	I	Noninverting signal input
OFFSET NULL	1, 5	I	Offset null pin used to eliminate the offset voltage and balance the input voltages.
OUTPUT	6	O	Amplified signal output
V+	7	I	Positive supply voltage
V-	4	I	Negative supply voltage

## 6.5 Electrical Characteristics, LM741<sup>(1)</sup>

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT
Input offset voltage	$R_B \leq 10 \text{ k}\Omega$ $T_A = 25^\circ\text{C}$ $T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$		1	3	mV
Input offset voltage adjustment range	$T_A = 25^\circ\text{C}$ , $V_S = \pm 20 \text{ V}$		$\pm 15$		mV
Input offset current	$T_A = 25^\circ\text{C}$ $T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$		20	200	nA
Input bias current	$T_A = 25^\circ\text{C}$ $T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$		85	500	nA
Input resistance	$T_A = 25^\circ\text{C}$ , $V_S = \pm 20 \text{ V}$		80	500	M $\Omega$
Input voltage range	$T_A = 25^\circ\text{C}$ , $V_S = \pm 20 \text{ V}$ $T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$	0.3	2		V
Large signal voltage gain	$V_S = \pm 15 \text{ V}$ , $V_O = \pm 10 \text{ V}$ , $R_L \geq 2 \text{ k}\Omega$ $T_A = 25^\circ\text{C}$ $T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$	50	200		V/mV
Output voltage swing	$V_S = \pm 15 \text{ V}$ $R_L \geq 10 \text{ k}\Omega$ $R_L \geq 2 \text{ k}\Omega$	$\pm 12$ $\pm 10$	$\pm 14$ $\pm 13$		V
Output short circuit current	$T_A = 25^\circ\text{C}$		25		mA
Common-mode rejection ratio	$R_B \leq 10 \text{ }\Omega$ , $V_{\text{CM}} = \pm 12 \text{ V}$ , $T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$	80	95		dB
Supply voltage rejection ratio	$V_S = \pm 20 \text{ V}$ to $V_S = \pm 5 \text{ V}$ , $R_B \leq 10 \text{ }\Omega$ , $T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$	86	96		dB
Transient response	Rise time Overshoot	$T_A = 25^\circ\text{C}$ , unity gain	0.3 5%		$\mu\text{s}$
Slew rate	$T_A = 25^\circ\text{C}$ , unity gain		0.5		V/ $\mu\text{s}$
Supply current	$T_A = 25^\circ\text{C}$		1.7	2.8	mA
Power consumption	$V_S = \pm 15 \text{ V}$ $T_A = 25^\circ\text{C}$ $T_A = T_{\text{MIN}}$ $T_A = T_{\text{MAX}}$		50 80 45	85 100 75	mW

(1) Unless otherwise specified, these specifications apply for  $V_S = \pm 15 \text{ V}$ ,  $-55^\circ\text{C} \leq T_A \leq +125^\circ\text{C}$  (LM741/LM741A). For the LM741C/LM741E, these specifications are limited to  $0^\circ\text{C} \leq T_A \leq +70^\circ\text{C}$ .

## 6.6 Electrical Characteristics, LM741A<sup>(1)</sup>

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT
Input offset voltage	$R_B \leq 50 \text{ }\Omega$ $T_A = 25^\circ\text{C}$ $T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$		0.8	3	mV
Average input offset voltage drift				4	mV/ $^\circ\text{C}$
Input offset voltage adjustment range	$T_A = 25^\circ\text{C}$ , $V_S = \pm 20 \text{ V}$		$\pm 10$		mV
Input offset current	$T_A = 25^\circ\text{C}$ $T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$		3	30	nA
Average input offset current drift				0.5	nA/ $^\circ\text{C}$
Input bias current	$T_A = 25^\circ\text{C}$ $T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$		30	80	nA
Input resistance	$T_A = 25^\circ\text{C}$ , $V_S = \pm 20 \text{ V}$ $T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$ , $V_S = \pm 20 \text{ V}$	1	8		M $\Omega$
Large signal voltage gain	$V_S = \pm 20 \text{ V}$ , $V_O = \pm 15 \text{ V}$ , $R_L \geq 2 \text{ k}\Omega$ $T_A = 25^\circ\text{C}$ $T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$	0.5 50 32			V/mV

(1) Unless otherwise specified, these specifications apply for  $V_S = \pm 15 \text{ V}$ ,  $-55^\circ\text{C} \leq T_A \leq +125^\circ\text{C}$  (LM741/LM741A). For the LM741C/LM741E, these specifications are limited to  $0^\circ\text{C} \leq T_A \leq +70^\circ\text{C}$ .



Electrical Characteristics, LM741A<sup>(1)</sup> (continued)

PARAMETER		TEST CONDITIONS		MIN	TYP	MAX	UNIT
Output voltage swing	$V_S = \pm 20\text{ V}$	$R_L \geq 10\text{ k}\Omega$		$\pm 16$			V
		$R_L \geq 2\text{ k}\Omega$		$\pm 15$			
Output short circuit current	$T_A = 25^\circ\text{C}$			10	25	35	mA
		$T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$		10		40	
Common-mode rejection ratio	$R_S \leq 50\ \Omega$ , $V_{\text{CM}} = \pm 12\text{ V}$ , $T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$			80	95		dB
Supply voltage rejection ratio	$V_S = \pm 20\text{ V}$ to $V_S = \pm 5\text{ V}$ , $R_S \leq 50\ \Omega$ , $T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$			86	96		dB
Transient response	Rise time	$T_A = 25^\circ\text{C}$ , unity gain		0.25	0.8		$\mu\text{s}$
	Overshoot			6%	20%		
Bandwidth <sup>(2)</sup>	$T_A = 25^\circ\text{C}$			0.437	1.5		MHz
Slew rate	$T_A = 25^\circ\text{C}$ , unity gain			0.3	0.7		V/ $\mu\text{s}$
Power consumption	$V_S = \pm 20\text{ V}$	$T_A = 25^\circ\text{C}$			80	150	mW
		$T_A = T_{\text{MIN}}$				165	
		$T_A = T_{\text{MAX}}$				135	

(2) Calculated value from: BW (MHz) = 0.35/Rise Time ( $\mu\text{s}$ )6.7 Electrical Characteristics, LM741C<sup>(1)</sup>

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT	
Input offset voltage	$R_S \leq 10\text{ k}\Omega$	$T_A = 25^\circ\text{C}$		2	6	mV	
		$T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$			7.5	mV	
Input offset voltage adjustment range	$T_A = 25^\circ\text{C}$ , $V_S = \pm 20\text{ V}$			$\pm 15$		mV	
Input offset current	$T_A = 25^\circ\text{C}$			20	200	nA	
		$T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$			300	nA	
Input bias current	$T_A = 25^\circ\text{C}$			80	500	nA	
		$T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$			0.8	$\mu\text{A}$	
Input resistance	$T_A = 25^\circ\text{C}$ , $V_S = \pm 20\text{ V}$		0.3	2		M $\Omega$	
Input voltage range	$T_A = 25^\circ\text{C}$		$\pm 12$	$\pm 13$		V	
Large signal voltage gain	$V_S = \pm 15\text{ V}$ , $V_O = \pm 10\text{ V}$ , $R_L \geq 2\text{ k}\Omega$	$T_A = 25^\circ\text{C}$		20	200	V/mV	
		$T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$		15			
Output voltage swing	$V_S = \pm 15\text{ V}$	$R_L \geq 10\text{ k}\Omega$	$\pm 12$	$\pm 14$		V	
		$R_L \geq 2\text{ k}\Omega$	$\pm 10$	$\pm 13$			
Output short circuit current	$T_A = 25^\circ\text{C}$			25		mA	
Common-mode rejection ratio	$R_S \leq 10\text{ k}\Omega$ , $V_{\text{CM}} = \pm 12\text{ V}$ , $T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$		70	90		dB	
Supply voltage rejection ratio	$V_S = \pm 20\text{ V}$ to $V_S = \pm 5\text{ V}$ , $R_S \leq 10\text{ }\Omega$ , $T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$		77	96		dB	
Transient response	Rise time	$T_A = 25^\circ\text{C}$ , Unity Gain		0.3		$\mu\text{s}$	
	Overshoot			5%			
Slew rate	$T_A = 25^\circ\text{C}$ , Unity Gain			0.5	V/ $\mu\text{s}$		
Supply current	$T_A = 25^\circ\text{C}$			1.7	2.8	mA	
Power consumption	$V_S = \pm 15\text{ V}$ , $T_A = 25^\circ\text{C}$			50	83	mW	

(1) Unless otherwise specified, these specifications apply for  $V_S = \pm 15\text{ V}$ ,  $-55^\circ\text{C} \leq T_A \leq +125^\circ\text{C}$  (LM741/LM741A). For the LM741C/LM741E, these specifications are limited to  $0^\circ\text{C} \leq T_A \leq +70^\circ\text{C}$ .

## 8 Application and Implementation

### NOTE

Information in the following applications sections is not part of the TI component specification, and TI does not warrant its accuracy or completeness. TI's customers are responsible for determining suitability of components for their purposes. Customers should validate and test their design implementation to confirm system functionality.

### 8.1 Application Information

The LM741 is a general-purpose amplifier that can be used in a variety of applications and configurations. One common configuration is in a noninverting amplifier configuration. In this configuration, the output signal is in phase with the input (not inverted as in the inverting amplifier configuration), the input impedance of the amplifier is high, and the output impedance is low. The characteristics of the input and output impedance is beneficial for applications that require isolation between the input and output. No significant loading will occur from the previous stage before the amplifier. The gain of the system is set accordingly so the output signal is a factor larger than the input signal.

### 8.2 Typical Application

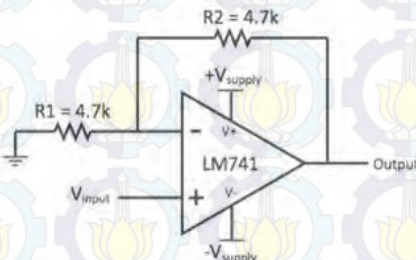


Figure 1. LM741 Noninverting Amplifier Circuit

#### 8.2.1 Design Requirements

As shown in [Figure 1](#), the signal is applied to the noninverting input of the LM741. The gain of the system is determined by the feedback resistor and input resistor connected to the inverting input. The gain can be calculated by [Equation 1](#):

$$\text{Gain} = 1 + (R2/R1) \quad (1)$$

The gain is set to 2 for this application. R1 and R2 are 4.7-k resistors with 5% tolerance.

#### 8.2.2 Detailed Design Procedure

The LM741 can be operated in either single supply or dual supply. This application is configured for dual supply with the supply rails at  $\pm 15$  V. The input signal is connected to a function generator. A 1-V<sub>pp</sub>, 10-kHz sine wave was used as the signal input. 5% tolerance resistors were used, but if the application requires an accurate gain response, use 1% tolerance resistors.

### 3. Datasheet IC4N35

#### 4N35, 4N36, 4N37 OPTOCOUPERS

SOE5021C—NOVEMBER 1981—REVISED APRIL 1998

#### COMPATIBLE WITH STANDARD TTL INTEGRATED CIRCUITS

- Gallium-Arsenide-Diode Infrared Source  
Optically Coupled to a Silicon npn  
Phototransistor
- High Direct-Current Transfer Ratio
- High-Voltage Electrical Isolation  
1.5-kV, 2.5-kV, or 3.55-kV Rating
- High-Speed Switching  
 $t_r = 7 \mu s$ ,  $t_f = 7 \mu s$  Typical
- Typical Applications Include Remote  
Terminal Isolation, SCR and Triac Triggers,  
Mechanical Relays and Pulse Transformers
- Safety Regulatory Approval  
UL/CUL, File No. E65085

#### DCJT OR 6-TERMINAL DUAL-IN-LINE PACKAGE

(TOP VIEW)



14N35 only  
NC—No internal connection

schematic



#### absolute maximum ratings at 25°C free-air temperature (unless otherwise noted)<sup>†</sup>

Input-to-output peak voltage (8-ms half sine wave):	4N35	3.55 kV
	4N36	2.5 kV
	4N37	1.5 kV
Input-to-output root-mean-square voltage (8-ms half sine wave):	4N35	2.5 kV
	4N36	1.75 kV
	4N37	1.05 kV
Collector-base voltage		70 V
Collector-emitter voltage (see Note 1)		30 V
Emitter-base voltage		7 V
Input-diode reverse voltage		6 V
Input-diode forward current:		60 mA
Peak (1 $\mu s$ , 300 pps)		3 A
Phototransistor continuous collector current		100 mA
Continuous total power dissipation at (or below) 25°C free-air temperature:		
Infrared-emitting diode (see Note 2)		100 mW
Phototransistor (see Note 3)		300 mW
Continuous power dissipation at (or below) 25°C lead temperature:		
Infrared-emitting diode (see Note 4)		100 mW
Phototransistor (see Note 5)		500 mW
Operating temperature range, $T_A$		–55°C to 100°C
Storage temperature range, $T_{stg}$		–55°C to 150°C
Lead temperature 1.6 mm (1/16 inch) from case for 10 seconds		260°C

<sup>†</sup> Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these conditions is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTES: 1. This value applies when the base-emitter diode is open-circuited.

2. Derate linearly to 100°C free-air temperature at the rate of 1.33 mW/°C.

3. Derate linearly to 100°C free-air temperature at the rate of 4 mW/°C.

4. Derate linearly to 100°C lead temperature at the rate of 1.33 mW/°C. Lead temperature is measured on the collector lead 0.8 mm (1/32 inch) from the case.

5. Derate linearly to 100°C lead temperature at the rate of 6.7 mW/°C.



Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

PRODUCTION DATA information is current as of publication date. Products conform to specifications at the time of their manufacture. Production processing does not necessarily include testing of all parameters.

TEXAS  
INSTRUMENTS

POST OFFICE BOX 655502 • DALLAS, TEXAS 75265

Copyright © 1998, Texas Instruments Incorporated

1



# 4N35, 4N36, 4N37 OPTOCOUPERS

SDS5021C - NOVEMBER 1981 - REVISED APRIL 1996

## electrical characteristics at 25°C free-air temperature (unless otherwise noted)

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT
$V_{BR(CBO)}$	Collector-base breakdown voltage	$I_C = 100 \mu A$ , $I_E = 0$ , $I_F = 0$	70 <sup>†</sup>		V
$V_{BR(CEO)}$	Collector-emitter breakdown voltage	$I_C = 10 \text{ mA}$ , $I_B = 0$ , $I_F = 0$	30 <sup>†</sup>		V
$V_{BR(ERO)}$	Emitter-base breakdown voltage	$I_E = 100 \mu A$ , $I_C = 0$ , $I_F = 0$	7 <sup>†</sup>		V
$I_R$	Input diode static reverse current	$V_R = 6 \text{ V}$		10 <sup>†</sup>	$\mu A$
$I_O$	Input-to-output current	$V_{IO} = \text{rated peak value}$ , $t = 8 \text{ ms}$		100	$\text{mA}$
$I_C(\text{on})$	On-state collector current	$V_{CE} = 10 \text{ V}$ , $I_F = 10 \text{ mA}$ , $I_B = 0$	10 <sup>†</sup>		$\text{mA}$
		$V_{CE} = 10 \text{ V}$ , $I_F = 10 \text{ mA}$ , $I_B = 0$ , $T_A = -55^\circ\text{C}$	4 <sup>†</sup>		
		$V_{CE} = 10 \text{ V}$ , $I_F = 10 \text{ mA}$ , $I_B = 0$ , $T_A = 100^\circ\text{C}$	4 <sup>†</sup>		
$I_C(\text{off})$	Off-state collector current	$V_{CE} = 10 \text{ V}$ , $I_F = 0$ , $I_B = 0$		1	$\text{nA}$
		$V_{CE} = 30 \text{ V}$ , $I_F = 0$ , $I_B = 0$ , $T_A = 100^\circ\text{C}$		500 <sup>†</sup>	
$h_{FE}$	Transistor static forward current transfer ratio	$V_{CE} = 5 \text{ V}$ , $I_C = 10 \text{ mA}$ , $I_F = 0$		500	
$V_F$	Input diode static forward voltage	$I_F = 10 \text{ mA}$	0.8 <sup>†</sup>	1.5 <sup>†</sup>	V
		$I_F = 10 \text{ mA}$ , $T_A = -55^\circ\text{C}$	0.6 <sup>†</sup>	1.7 <sup>†</sup>	
		$I_F = 10 \text{ mA}$ , $T_A = 100^\circ\text{C}$	0.7 <sup>†</sup>	1.4 <sup>†</sup>	
$V_{CE(\text{sat})}$	Collector-emitter saturation voltage	$I_C = 0.5 \text{ mA}$ , $I_F = 10 \text{ mA}$ , $I_B = 0 \text{ mA}$		0.3 <sup>†</sup>	V
$r_{IO}$	Input-to-output internal resistance	$V_{IO} = 500 \text{ V}$	10 <sup>11</sup>		$\Omega$
$C_{IO}$	Input-to-output capacitance	$V_{IO} = 0$ , $f = 1 \text{ MHz}$ , See Note 6		1	2.5 <sup>†</sup> $\text{pF}$

† JEDEC registered data

NOTE 6: These parameters are measured between both input-diode leads shorted together and all the phototransistor leads shorted together.

## switching characteristics at 25°C free-air temperature<sup>†</sup>

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT
$t_{\text{on}}$	Time-on time	$V_{CC} = 10 \text{ V}$ , $I_C(\text{on}) = 2 \text{ mA}$	7	10	$\mu\text{s}$
$t_{\text{off}}$	Turn-off time	$R_L = 100 \Omega$ , See Figure 1	7	10	

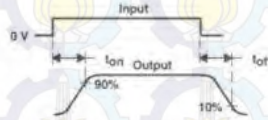
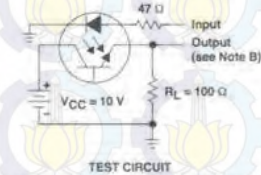
† JEDEC registered data



# 4N35, 4N36, 4N37 OPTOCOUPLEDERS

SOE5021C - NOVEMBER 1981 - REVISED APRIL 1998

## PARAMETER MEASUREMENT INFORMATION

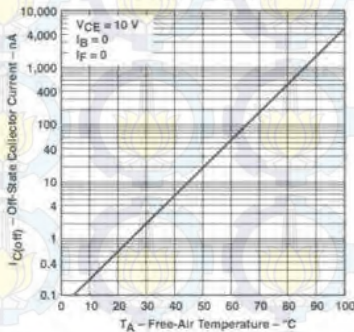


- NOTES: A. The input waveform is supplied by a generator with the following characteristics:  $Z_0 = 50 \Omega$ ,  $t_r \leq 15$  ns, duty cycle = 1%,  $t_w = 100$   $\mu$ s.  
B. The output waveform is monitored on an oscilloscope with the following characteristics:  $t_r \leq 12$  ns,  $R_{in} \approx 1$  M $\Omega$ ,  $C_{in} \approx 20$  pF.

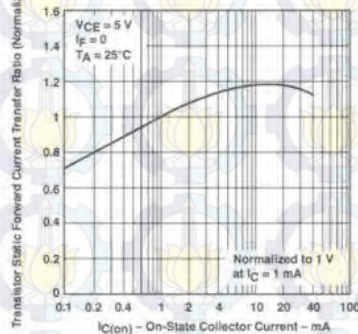
Figure 1. Switching Times

## TYPICAL CHARACTERISTICS

### OFF-STATE COLLECTOR CURRENT vs FREE-AIR TEMPERATURE



### TRANSISTOR STATIC FORWARD CURRENT TRANSFER RATIO (NORMALIZED) vs ON-STATE COLLECTOR CURRENT



## 4. Datasheet MOC3021

### MOC3020 THRU MOC3023 OPTOCOUPERS/OPTOISOLATORS

50ES025A - OCTOBER 1988 - REVISED APRIL 1998

- 400 V Phototriac Driver Output
- Gallium-Arsenide-Diode Infrared Source and Optically-Coupled Silicon Triac Driver (Bilateral Switch)
- UL Recognized . . . File Number E65085
- High Isolation . . . 7500 V Peak
- Output Driver Designed for 220 Vac
- Standard 6-Terminal Plastic DIP
- Directly Interchangeable with Motorola MOC3020, MOC3021, MOC3022, and MOC3023

typical 115/240 Vac(rms) applications

- Solenoid/Valve Controls
- Lamp Ballasts
- Interfacing Microprocessors to 115/240 Vac Peripherals
- Motor Controls
- Incandescent Lamp Dimmers

#### absolute maximum ratings at 25°C free-air temperature (unless otherwise noted)†

Input-to-output peak voltage, 5 s maximum duration, 60 Hz (see Note 1)	7.5 kV
Input diode reverse voltage	3 V
Input diode forward current, continuous	50 mA
Output repetitive peak off-state voltage	400 V
Output on-state current, total rms value (50-60 Hz, full sine wave): $T_A = 25^\circ\text{C}$	100 mA
$T_A = 70^\circ\text{C}$	50 mA
Output driver nonrepetitive peak on-state current ( $t_{on} = 10$ ms, duty cycle = 10%, see Figure 7)	1.2 A
Continuous power dissipation at (or below) 25°C free-air temperature:	
Infrared-emitting diode (see Note 2)	100 mW
Phototriac (see Note 3)	300 mW
Total device (see Note 4)	330 mW
Operating junction temperature range, $T_J$	-40°C to 100°C
Storage temperature range, $T_{STG}$	-40°C to 150°C
Lead temperature 1/6 (1/16 inch) from case for 10 seconds	260°C

† Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTES: 1. Input-to-output peak voltage is the internal device dielectric breakdown rating.

2. Derate linearly to 100°C free-air temperature at the rate of 1.33 mW/°C.

3. Derate linearly to 100°C free-air temperature at the rate of 4 mW/°C.

4. Derate linearly to 100°C free-air temperature at the rate of 4.4 mW/°C.

MOC3020 - MOC3023 . . . PACKAGE  
(TOP VIEW)



† Do not connect this terminal  
NC - No internal connection

logic diagram



PRODUCTION DATA information is current as of publication date.  
Products conform to specifications on the basis of Texas Instruments  
standard warranty. Production processing does not necessarily include  
testing of all parameters.



TEXAS  
INSTRUMENTS

POST OFFICE BOX 655508 • DALLAS, TEXAS 75265

Copyright © 1998, Texas Instruments Incorporated

# MOC3020 THRU MOC3023 OPTOCOUPERS/OPTOISOLATORS

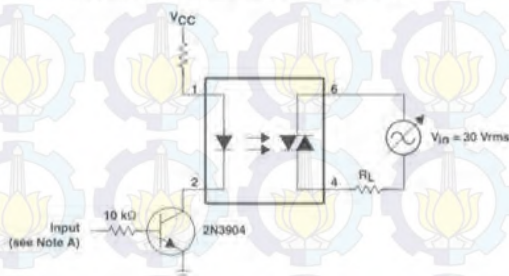
SOE3025A - OCTOBER 1986 - REVISED APRIL 1998

electrical characteristics at 25°C free-air temperature (unless otherwise noted)

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
$I_R$	Static reverse current	$V_R = 3 \text{ V}$		0.05	100	$\mu\text{A}$
$V_F$	Static forward voltage	$I_F = 10 \text{ mA}$		1.2	1.5	V
$I_{(PRM)}$	Repetitive off-state current, either direction	$V_{(PRM)} = 400 \text{ V}$ , See Note 5		10	100	nA
$dv/dt$	Critical rate of rise of off-state voltage	See Figure 1		100		V/ $\mu\text{s}$
$dv/dt(c)$	Critical rate of rise of commutating voltage	$I_Q = 15 \text{ mA}$ , See Figure 1		0.15		V/ $\mu\text{s}$
$I_{FT}$	Input trigger current, either direction	MOC3020		15	30	mA
		MOC3021		8	15	
		MOC3022		5	10	
		MOC3023		3	5	
$V_{TM}$	Peak on-state voltage, either direction	$I_{RM} = 100 \text{ mA}$		1.4	3	V
$I_H$	Holding current, either direction			100		$\mu\text{A}$

NOTE 5: Test voltage must be applied at a rate no higher than 12 V/ $\mu\text{s}$ .

## PARAMETER MEASUREMENT INFORMATION



NOTE A: The critical rate of rise of off-state voltage,  $dv/dt$ , is measured with the input at 0 V. The frequency of  $V_{IN}$  is increased until the phototriac turns on. This frequency is then used to calculate the  $dv/dt$  according to the formula:

$$dv/dt = 2 \cdot \sqrt{2} \cdot \pi \cdot V_{IN} \cdot f$$

The critical rate of rise of commutating voltage,  $dv/dt(c)$ , is measured by applying occasional 5-V pulses to the input and increasing the frequency of  $V_{IN}$  until the phototriac stays on (latches) after the input pulse has ceased. With no further input pulses, the frequency of  $V_{IN}$  is then gradually decreased until the phototriac turns off. The frequency at which turn-off occurs may then be used to calculate the  $dv/dt(c)$  according to the formula shown above.

Figure 1. Critical Rate of Rise Test Circuit



# MOC3020 THRU MOC3023 OPTOCOUPLEDERS/OPTOISOLATORS

SD/SG25A - OCTOBER 1986 - REVISED APRIL 1998

## APPLICATIONS INFORMATION

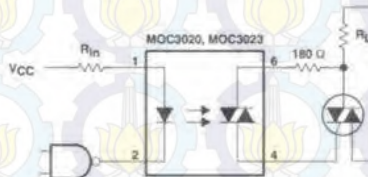


Figure 5. Resistive Load

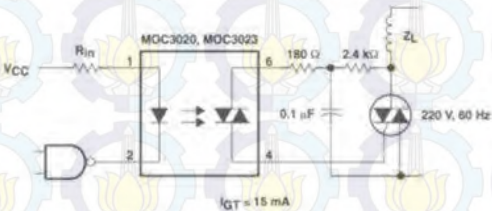


Figure 6. Inductive Load With Sensitive-Gate Triac

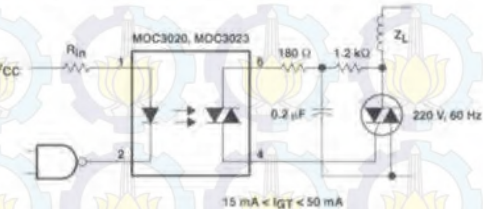


Figure 7. Inductive Load With Nonsensitive-Gate Triac



## 5. Datasheet TRIAC BTA16

### BTA16-600CW3G, BTA16-800CW3G

#### Triacs

#### Silicon Bidirectional Thyristors

Designed for high performance full-wave ac control applications where high noise immunity and high commutating di/dt are required.

#### Features

- Blocking Voltage to 800 V
- On-State Current Rating of 16 A RMS at 25°C
- Uniform Gate Trigger Currents in Three Quadrants
- High Immunity to dV/dt – 1000 V/μs minimum at 125°C
- Minimizes Snubber Networks for Protection
- Industry Standard TO-220AB Package
- High Commutating di/dt – 8.5 A/ms minimum at 125°C
- Internally Isolated (2500 V<sub>RMS</sub>)
- These Devices are Pb-Free and are RoHS Compliant\*

#### MAXIMUM RATINGS (T<sub>J</sub> = 25°C unless otherwise noted)

Rating	Symbol	Value	Unit
Peak Repetitive Off-State Voltage (Note 1) (T <sub>J</sub> = -40 to 125°C, Sine Wave, 50 to 60 Hz, Gate Open)	V <sub>ORM</sub> V <sub>RRM</sub>	600 800	V
On-State RMS Current (Full Cycle Sine Wave, 60 Hz, T <sub>C</sub> = 25°C)	I <sub>T(RMS)</sub>	16	A
Peak Non-Repetitive Surge Current (One Full Cycle Sine Wave, 60 Hz, T <sub>C</sub> = 25°C)	I <sub>TSM</sub>	170	A
Circuit Fusing Consideration (t = 10 ms)	I <sub>FT</sub>	144	A <sup>2</sup> sec
Non-Repetitive Surge Peak Off-State Voltage (T <sub>J</sub> = 25°C, t = 10ms)	V <sub>DSM</sub> V <sub>RRSM</sub>	V <sub>DSM</sub> /V <sub>RRSM</sub> +100	V
Peak Gate Current (T <sub>J</sub> = 125°C, t = 20 μs)	I <sub>GM</sub>	4.0	A
Peak Gate Power (Pulse Width ≤ 1.0 μs, T <sub>C</sub> = 80°C)	P <sub>GM</sub>	20	W
Average Gate Power (T <sub>J</sub> = 125°C)	P <sub>G(AV)</sub>	1.0	W
Operating Junction Temperature Range	T <sub>J</sub>	-40 to +125	°C
Storage Temperature Range	T <sub>stg</sub>	-40 to +150	°C
RMS Isolation Voltage (t = 300 ms, R.H. ≤ 30%, T <sub>A</sub> = 25°C)	V <sub>IS</sub>	2500	V

Stresses exceeding Maximum Ratings may damage the device. Maximum Ratings are stress ratings only. Functional operation above the Recommended Operating Conditions is not implied. Extended exposure to stresses above the Recommended Operating Conditions may affect device reliability.

1. V<sub>ORM</sub> and V<sub>RRM</sub> for all types can be applied on a continuous basis. Blocking voltages shall not be tested with a constant current source such that the voltage ratings of the devices are exceeded.

\*For additional information on our Pb-Free strategy and soldering details, please download the ON Semiconductor Soldering and Mounting Techniques Reference Manual, SOLDERRM/D.

ON Semiconductor®

<http://onsemi.com>

**TRIACS**  
**16 AMPERES RMS**  
**600 thru 800 VOLTS**



TO-220AB  
CASE 221A  
STYLE 12

#### MARKING DIAGRAM



x = 6 or 8  
A = Assembly Location (Optional)\*  
Y = Year  
WW = Work Week  
G = Pb-Free Package

\*The Assembly Location code (A) is optional. In cases where the Assembly Location is stamped on the package the assembly code may be blank.

#### PIN ASSIGNMENT

1	Main Terminal 1
2	Main Terminal 2
3	Gate
4	No Connection

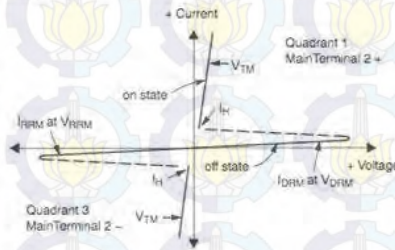
#### ORDERING INFORMATION

Device	Package	Shipping
BTA16-600CW3G	TO-220AB (Pb-Free)	50 Units / Rail
BTA16-800CW3G	TO-220AB (Pb-Free)	50 Units / Rail

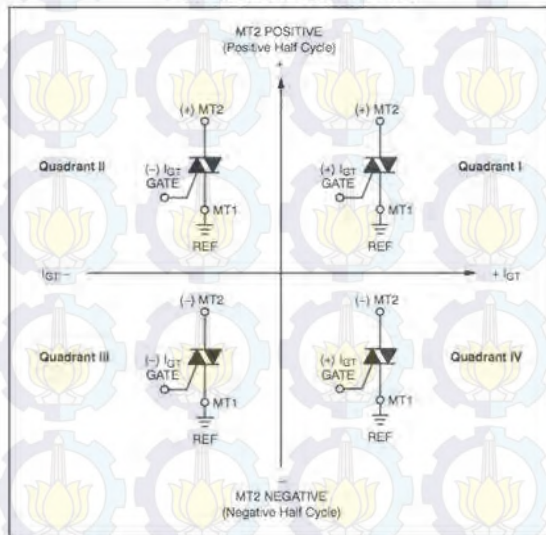
## BTA16-600CW3G, BTA16-800CW3G

### Voltage Current Characteristic of Triacs (Bidirectional Device)

Symbol	Parameter
$V_{DRM}$	Peak Repetitive Forward Off State Voltage
$I_{PRM}$	Peak Forward Blocking Current
$V_{RRM}$	Peak Repetitive Reverse Off State Voltage
$I_{RRM}$	Peak Reverse Blocking Current
$V_{TM}$	Maximum On State Voltage
$I_H$	Holding Current

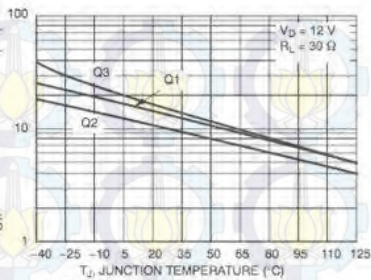


### Quadrant Definitions for a Triac

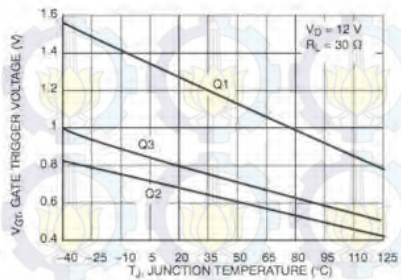


All polarities are referenced to MT1.  
With in-phase signals (using standard AC lines) quadrants I and III are used.

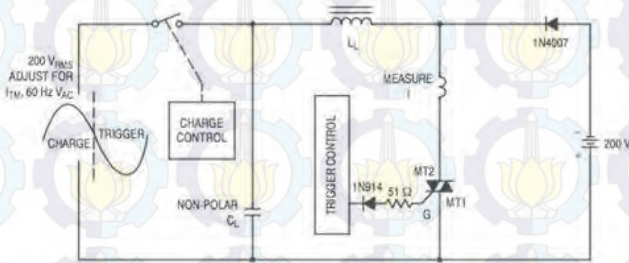
# **BTA16-600CW3G, BTA16-800CW3G**



**Figure 6. Gate Trigger Current Variation**



**Figure 7. Gate Trigger Voltage Variation**



Note: Component values are for verification of rated  $(di/dt)_c$ . See AN1048 for additional information.

**Figure 8. Simplified Test Circuit to Measure the Critical Rate of Rise of Commutating Current  $(di/dt)_c$**





## DAFTAR RIWAYAT HIDUP



Nama : Ratih Putri Nastiti  
TTL : Cepu, 19 Nopember 1995  
Jenis Kelamin : Perempuan  
Agama : Islam  
Alamat : Jalan Dumai no 85,  
Nglajo, Cepu, Blora, Jawa  
Tengah  
Telp/HP : 082245313364  
E-mail : rpnastiti@gmail.com

### RIWAYAT PENDIDIKAN

1. 2000 – 2006 : SD Negeri XIV Cepu
2. 2006 – 2009 : SMP Negeri 3 Cepu
3. 2009 – 2012 : SMA Negeri 1 Semarang
4. 2012 – 2015 : D3 Teknik Elektro, Program Studi Teknik Elektro Industri - FTI Institut Teknologi Sepuluh Nopember (ITS)

### PENGALAMAN KERJA

1. Kerja Praktek di Pusdiklat Migas Cepu
2. Kerja Praktek di PJB Gresik

### PENGALAMAN ORGANISASI

1. Staff Kewirausahaan Periode 2014/2015 HIMAD3TEKTRO, FTI - ITS
2. Staff Konsep Kreatif ITS EXPO Periode 2014/2015 HIMAD3TEKTRO, FTI - ITS

## DAFTAR RIWAYAT HIDUP



Nama : Wahyu Susanto  
TTL : Surabaya, 14 Juli 1994  
Jenis Kelamin : Laki - Laki  
Agama : Islam  
Alamat : Jalan Kejawan Gebang,  
Kel. Gebang Putih, Kec.  
Sukolilo, Surabaya  
Telp/HP : 083830462233  
E-mail : wahyususanto800@gmail.  
com

### RIWAYAT PENDIDIKAN

1. 2000 – 2006 : SD YAPITA Surabaya
2. 2006 – 2009 : SMP Negeri 19 Surabaya
3. 2009 – 2013 : SMK Negeri 5 Surabaya
4. 2013 – 2016 : D3 Teknik Elektro, Program Studi Teknik Elektro Industri - FTI Institut Teknologi Sepuluh Nopember (ITS)

### PENGALAMAN KERJA

1. Kerja Praktek di PT. PAL Indonesia (persero), Surabaya
2. Kerja Praktek di PT. PJB UP Gresik

### PENGALAMAN ORGANISASI

1. Anggota Staff HUBLU Himpunan Mahasiswa D3 Teknik Elektro 2014/2015